DOCUMENT ANALYSIS AND SCRIPT IDENTIFICATION

BY

ZAHER AHMED SAHAL BAMASOOD

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER SCIENCE

October 2013

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **ZAHER AHMED SAHAL BAMASOOD** under the direction his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER SCIENCE.**

Dr. Sabri A. Mahmoud
(Advisor)

Dr. Adel F. Ahmed
Department Chairman

Dr. Radwan E. Abdel-Aal
(Member)

Dr. Salam A. Zummo
Dean of Graduate Studies

Dr. Wasfi G. Al-Khatib
(Member)

22/12/13
Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

DEDICATION

I dedicate this work to all my family members, especially my parents, my wife and my children, Ahmed and Jna.

ACKNOWLEDGMENTS

I would like to thank my supervisor, Dr. Sabri Mahmoud, for having helped me realize this research and for guidance during the course of this work. I also thank him for the time and effort he put into reading many drafts of this thesis and for his extensive and valuable advice.

Special thanks to the other members of my thesis committee Dr. Radwan Abdel-Aal and Dr. Wasfi Al-Khatib.

I would like to acknowledge the partial support provided by King Abdul-Aziz City for Science and Technology (KACST), under project no. AT – 30 – 53, through King Fahd University of Petroleum & Minerals (KFUPM).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Full Name: ZAHER AHMED SAHAL BAMASOOD

Thesis Title: DOCUMENT ANALYSIS AND SCRIPT IDENTIFICATION

Major Field: COMPUTER SCIENCE

Date of Degree: October 2013

The main objective of this thesis is developing techniques to segment a document into text and non text regions, then classify the scripts of each text region as Arabic or Latin. The system is divided into document analysis and classification, and script identification. In Document analysis and classification, we proposed an algorithm to segment a document into homogenous regions that are later classified into text or non text. The database is 398 images collected from Printed Arabic Text Database (PATDB), which includes 6074 regions partitioned into 4231 for training and 1843 for testing. The proposed algorithm has the best performance in merged zones error when it was compared with XY cut and RLSA segmentation algorithms. We used Sequential Forward Features Selection (SFFS) and Sequential Backward Features Selection (SBFS) to select the best features. The best features are evaluated by using neural network (NN), SVM, and K-NN (K=1) classifiers. The proposed algorithm shows the best performance in all cases except the case of using SBFS method with NN classifier. In script identification, Gabor features are extracted at block and word levels. A database of 444 pages was collected from PATDB, University of Washington and our own. The scripts are identified at the block and word levels using K-NN (K=1), Nearest Mean (NM), NN, SVM, Decision Tree, and Tree Boost classifiers. SVM shows the highest accuracy with 99.5952%, and 99.76% at the block and word levels, respectively. NM shows the lowest accuracy at the block and word levels.

ملخص الرسالة

الاسم الكامل: زاهر أحمد سهل بامسعود

عنوان الرسالة: تحليل المستند وتحديد اللغة

التخصص: علوم الحاسب الآلي.

تاريخ الدرجة العلمية: أكتوبر 2013.

الهدف الأساسي من هذه الرسالة هو تطوير تقنيات تقوم بتقسيم مستند إلى مناطق نصية وغير نصية، ثم تصنيف لغات المناطق النصية إلى عربي أو لاتيني. ينقسم النظام إلى جزأين: تحليل وتصنيف المستند، وتحديد اللغة. اقترحنا خوارزمية في تحليل وتصنيف المستند لتقسيم مستند إلى مناطق متجانسة، وثم تصنف المناطق المتجانسة إلى نصية أو غير نصية. تتكون قاعدة البيانات من 398 صورة مأخودة من قاعدة النص العربي المطبوع (المقدمة في جامعة الملك فهد للبترول والمعادن) والتي تشمل 6074 منطقة مقسمة إلى 4231 منطقة للتدريب و1843 منطقة للفحص. أظهرت خوارزميتنا المقترحة بأنها الأفضل أداءً في مقياس الخطأ للمناطق المدموجة عندما قارناها بالخوارزميتين XY cut، وRLSA. لتحديد أفضل السمات اتبعنا طريقتين الأولى طريقة تحديد السمات بالتسلسل الأمامي والثانية طريقة تحديد السمات بالتسلسل الخلفي. لتقييم أفضل السمات، استخدمنا مصنفات الشبكة العصبية (NN)، والدعم الموجه الآلي (SVM) والجار الأقرب (K-NN). أظهرت خوارزميتنا المقترحة بأنها الأفضل في كل الحالات إلا في حالة تحديد السمات بالتسلسل الخلفي مع استخدام مصنف الشبكة العصبية.

وفي تحديد اللغة، استخرجنا سمات جابور على مستوى المنطقة والكلمة. تتكون قاعدة البيانات من 444 صورة مأخوذة من قاعدة النص العربي المطبوع، وقاعدة جامعة واشنطن الإصدار الأول، وقاعدة البيانات الخاصة بنا. المصنفات المستخدمة في تحديد اللغة على مستوى المنطقة والكلمة هي الجار الأقرب (K-NN)، والمتوسط الأقرب (Nearest Mean)، والشبكة العصبية (NN)، والدعم الموجه الآلي (SVM)، وشجرة القرار (Decision Tree)، وشجرة الدعم (Tree Boost). حيث اظهر مصنف الدعم الموجه الآلي النتائج الأفضل: وهي 99.5952% على مستوى المنطقة و99.76% على مستوى الكلمة. بينما أظهر مصنف المتوسط الأقرب النتائج الأقل على مستوى المنطقة والكلمة.

CHAPTER 1

INTRODUCTION

Document analysis and script identification play an important role in document image processing and its applications. Document analysis and script identification usually select the main directions of the whole process of information conversion to digital form. For documents containing text and image regions, document analysis and classification groups the contents of the document into text and image regions. Then other systems may be applied such as an optical character recognition (OCR) system to recognize text, and an image processing system to process the images. Most OCR systems can read characters written in one particular script only. Script identification is needed to separate multiple scripts in the text regions. An appropriate OCR system is then chosen for each script.

Document analysis and classification systems usually include document segmentation, feature extraction and document classification. In addition, document analysis and classification systems may require a preprocessing phase to enhance the document images by removing noise, correcting image skew, etc. Document segmentation divides a document into homogeneous regions. Because each region has its own characteristics, a number of features are extracted for each region. Finally, document classification uses one or more classifiers to classify each region based on the extracted features into text, mathematical formulas, tables, figures, etc.

Script identification is an important step in document images processing for multilingual documents. Scripts must be known early in order to choose an appropriate OCR system. Existing script identification methods usually extract various features from the document at different levels, such as page, text block, text line, word, and even character levels. Page level script identification methods assume one script on the whole page. In text block level methods, each text block (or text region) is assumed to have one script. Text line level script techniques assume one script for each line. At the word and character levels each text line can contain multi-scripts. In this research, the features are extracted from the page at the block and word levels.

This research has two main parts, namely 1) document analysis and classification, and 2) script identification. Document analysis and classification is divided into four phases: 1) preprocessing, 2) document segmentation, 3) feature extraction, and 4) document classification. Preprocessing is used to enhance the images by removing noise, correcting any image skew, etc. Document segmentation phase segments document image into homogeneous regions. In feature extraction, features are extracted for each region. Based on these features, we classify the regions in the document into text and non-text regions. Finally, the script identification classifies the scripts of the text regions as Arabic or Latin using Gabor filters.

1.1. Motivation

The development of a document analysis and classification system is necessary for several IT applications. For example, in OCR it is necessary to separate text regions from other regions to convert the text areas into editable text. The documents are constructed by replacing the text regions, which are in image format, into editable text. Then, to keep

the original structure of the document, the other regions are kept in place. In document retrieval and indexing applications, instead of searching for the image document, we can search the textual regions. Then the result of search will be faster with higher retrieval accuracy rates

Many document analysis and classification researches for different languages have been proposed. While document analysis and classification techniques for Arabic documents have received less attention.

A large number of Arabic documents (such as books, magazines, letters etc) are bilingual documents, their text regions can have English words. Although OCR field has received many researches for different languages, most OCR systems can recognize characters written only in one specific script. For that reasons, we need to identify script early in order to apply the appropriate OCR for each script.

1.2. Problem Description

Document analysis and script identification are significant fields in our life. Many researches and techniques in document analysis have been developed in the literature for different languages. However, Arabic has received less interest. In general, document analysis and classification has three important parts: document segmentation, features extraction, and classification. Document segmentation divides a document into homogeneous regions, while classification labels those regions as text, halftones, line drawings, etc. by using the features extracted from each region. Since some images may get distorted as a result of scanning, preprocessing is needed as an initial step to improve image quality and remove possible distortions.

This work addressed the problem of documents analysis and script identification. In addition, we provide implementation of documents analysis and script identification prototype which includes preprocessing, features extraction, document segmentation, documents classification, and script identification. A number of features are extracted from documents, then by using these features the image contents are classified into text, and non-text regions. Then the script of the text regions are classified to Arabic or Latin.

1.3. Thesis Objectives

The main objectives of this research are as follows:

1. Conduct research on document analysis and script identification and develop required techniques and algorithms.
2. Implement a document analysis and script identification prototype.
3. Organize an existing database that will be used in this research, and add more documents to the database for the work of this thesis.
4. Disseminate the outcomes of this research by writing the thesis and possibly publishing papers and data on this research topic.

1.4. Thesis Organization

The remaining chapters of this thesis are organized as follows. Chapter 2 provides background and literature review of document analysis and script identification. Chapter 3 presents and discusses the proposed document analysis and classification. Script identification is presented in chapter 4. Chapter 5 describes the experimental results, whereas chapter 6 concludes this work.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

Document Analysis and script identification research has been addressed by many researchers. There are many methods proposed in the literature to deal with Document analysis and classification. This literature review is organized into preprocessing, document analysis, and script identification.

2.1. Preprocessing

When a document paper is scanned then digitized, some noise result and skew may be present if the paper is not placed on the scanner correctly. For that reason, preprocessing is needed to enhance document images before segmentation. The preprocessing includes noise removal, binarization, skew detection and correction, etc.

Binarization is converting a gray-level image into a binary image. Image pixels are clustered into two groups, foreground and background. Each pixel on the image is transformed into black (background) or white (forground) based on its gray level. If its gray level is smaller than a threshold (T), it will be changed into white otherwise black. The Otsu algorithm [Otsu79] computes the optimal threshold for binarizing an image based on the information of its gray levels. The threshold is computed as the luminance level T that minimizes the weighted intra-group (or within) variance between foreground (F) and Background (B) or, equivalently, maximizes the corresponding inter-group (or between) variance. Equation (2.1) represents the intra-group variance while equation (2.2) represents the inter-group variance equation.

$$\sigma^2_{within}(T) = w_B(T)\sigma^2_B(T) + w_F(T)\sigma^2_F(T) \qquad (2.1)$$

$$\sigma^2_{between}(T) = w_B(T)w_F(T)\,[\mu_B(T) - \mu_F(T)]^2 \qquad (2.2)$$

Where $\sigma^2_B(T)$, and $\sigma^2_F(T)$ are the variance of pixels in the background and foreground respectively, $\mu_B(T)$, and $\mu_F(T)$ are the means of B and F determined by T. The weights ($w_B(T)$, and $w_F(T)$)are specific for each threshold, and are taken to be the class probability of B and F over the whole set of pixels.

For each gray level T, the ratio P(T) of pixels in the image having that gray level over the total number of pixels in the image (i.e., the probability of a pixel in the image having that gray level).Then, the probability distributions are:

$$w_F(T) = \sum_{i \in F} P(i)\,, \qquad w_B(T) = \sum_{i \in B} P(i) = 1 - w_F(T)$$

After computing the initial values $w_F(0)$, $w_B(0)$, $\mu_F(0)$, and $\mu_B(0)$, the next values can be obtained as:

$$w_F(T+1) = w_F(T) + P(T), \qquad w_B(T+1) = w_B(T) - P(T) = 1 - w_F(T+1)$$

$$\mu_F(T+1) = \frac{\mu_F(T)w_F(T) + P(T).T}{w_F(T+1)}, \qquad \mu_B(T+1) = \frac{\mu_B(T)w_B(T) + P(T).T}{w_B(T+1)}$$

Statistical-Based Smoothing algorithm [Mahm94] is used to remove the noise in an image. The main idea of this algorithm is that each pixel in the image is eliminated or filled based on its initial value and its neighbors' initial values. The rules are stated as follow:

If $P_0 = 0$

$$P_0' = \begin{cases} 0, & \text{if } \sum_{i=1}^{8} P_i < T \\ 1, & \text{otherwise} \end{cases}$$

Else

$$P_0' = \begin{cases} 1, & \text{if } P_i + P_{i+1} = 2 \text{ for at least one } i = 1, \dots, 8 \\ 0, & \text{otherwise} \end{cases}$$

Where P_0 is the current pixel, P_0' is the new value and T is a threshold. In the above rules, 0 represents white pixel, and 1 represents black. The labeling schemes of those pixels are shown in the Figure 1.

P_4	P_3	P_2
P_5	P_0	P_1
P_6	P_7	P_8

Figure 1: the current pixel and its neighbors.

A number of methods have been proposed in the literature to detect and correct skew angles for document images. These methods are categorized based on their techniques such as projection profiles, nearest neighbor clustering, piecewise covering by parallelograms, Hough transform, etc.

The main idea of the research presented in [Bagd97], [Post86], [Nicc99], [Bair92] is in two main steps: 1) creating horizontal projection profiles of a document image at various angles, and 2) the skew angle is detected where the maximum value of the projection is achieved. In [Bair92] the algorithm applies a connected component analysis to the document. Then it projects the midpoint of the bottom of each connected component onto

an imaginary accumulator line perpendicular to different projection angles. For each projection direction, the sum of squares of the accumulated values of each bin is computed. During the projection, large connected components are ignored. Then skew angle is selected based on the maximum value.

Some researchers used the nearest neighbor (NN) clustering technique to calculate skew angles [Gorm93], [Hash86]. In these techniques, the center of each connected component is connected with the center of its nearest neighbor, then the histogram of the nearest neighbor connection angles is taken to determine the skew angle. To our knowledge, the first method based on the nearest neighbor method was proposed by Hashizume et al [Hash86]. The direction vector of all nearest neighbor pairs of connected components are collected in a histogram and the dominant skew is determined by the peak in the histogram. Gorman extended the method to be K-nearest neighbors for each connected component [Gorm93].

Chou et al. proposed a skew detection method for scanned documents that estimates skew angles based on piecewise covering of items, such as text-lines, figures, forms, or tables [Chou07]. This method starts by drawing parallel lines (scan line) at some angles from left to right and dividing a document image vertically into a number of non-overlapping regions, called slabs. Each line is divided into sections where each section is defined as the part of a scan line that lies within a slab (see Figure 2.a). The parallelograms construction phase then starts by examining each section at certain angles. If this section contains at least one black pixel, it is changed to gray; otherwise, it stays white. Then count the number of white sections of the scan lines (B) and repeat the above operation

with drawing the lines at different angles. Finally, the estimated skew angle is θ, if B (θ) is the largest. Figure 2 shows the method.

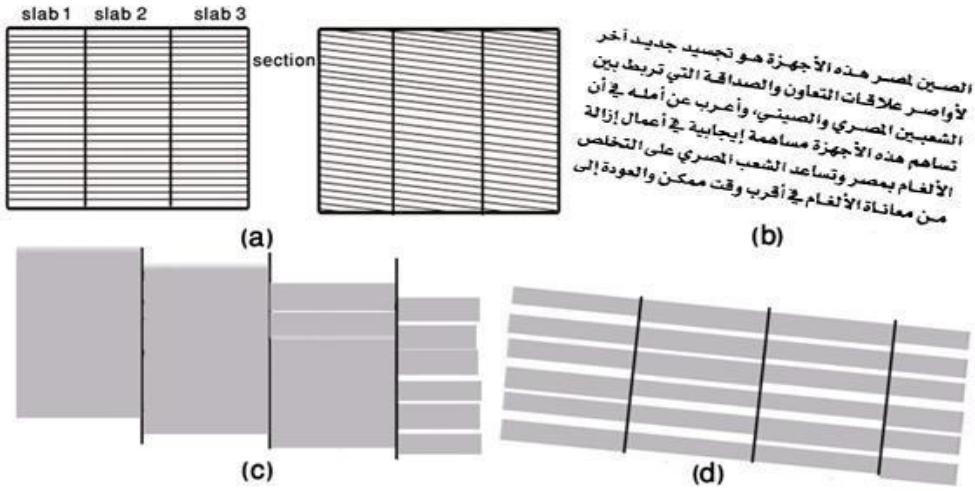


Figure 2: Chou et al.'s method (a) Parallel scan lines with skew angles 0°, and 6°; (b) Original image; (c) Parallelograms constructed at skew angle 0°; (d) Parallelograms constructed at skew angle at 6° [Chou07]

Some researchers used Hough transform to detect the skew angle, such as [YuJa96], [Manj07], [Srih89], [ChaC06]. In Hough transformation, a set of points in Cartesian space (x, y) are mapped to sinusoidal curves in the Hough space (ρ, θ) via the following transform:

$$\rho = x \cos\theta + y \sin\theta, \qquad \text{where } -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$$

Every time a sinusoidal curve intersects another curve at a particular value of ρ and θ, the likelihood increases that a line corresponding to that (ρ, θ) coordinates value is present in the original image. Figure 3 shows the Hough transform equation. An accumulator array is used to count the number of intersections of the various ρ and θ values. The skew is then determined by the θ values corresponding to the highest number of counts in the accumulator array.

A number of strategies were presented to make Hough transform technique faster by reducing the number of image pixels. In [YuJa96], the number of the pixels are reduced by considering the centroid of the connected components instead of processing all image pixels in the first stage (called the block adjacency graph). In [Manj07], the connected components from images are selected and blocked. Then the results are thinned to reduce image pixels. Finally, the thinned results are fed to the Hough transform

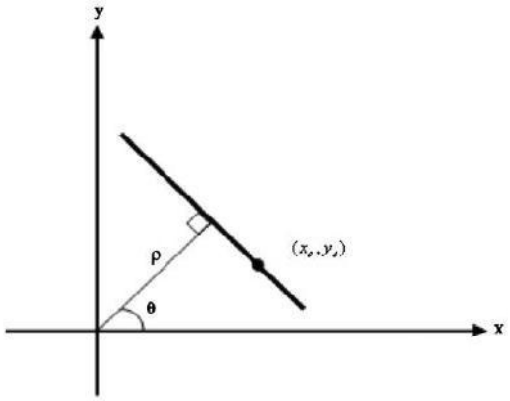


Figure 3: Polar form of the Hough Transform for the line, $\rho = x\cos\theta + y\sin\theta$

2.2. Document Analysis

In document analysis, a document is segmented into homogeneous regions in the document segmentation phase. Then in the feature extraction and classification phases a number of features are extracted from each region to classify it into text, table, figure, and etc.

2.2.1. Document Segmentation

To segment document images into homogenous regions, several algorithms have been proposed. Wong et. al. presented the run-length smoothing algorithm (RLSA) which is a bottom-up approach [Wong82]. This algorithm consists of three steps (see Figure 5). In the first step, horizontal white runs that are smaller or equal to a threshold are

changed to black runs. The second step is like the first but vertically not horizontally. The last step combines the results from the first and the second steps by the logical "and" operator.

Gorman proposed a bottom-up approach based on nearest-neighborhood clustering [Gorm93]. The main idea of this method is using a graph to connect each connected component to its k-nearest neighbors. So, if two characters in the same word form a nearest neighbor pair whose distance is relatively small and whose angle is close to zero, they will be connected. Most of the connections will be between characters on the same lines. The connections in each text line are then grouped with the underlying connections to create blocks or regions (see Figure 4).

X-Y cut algorithm which is a top-down algorithm was presented in [Jaek95], [Nagy92]. In this algorithm, the page must be unskewed and the homogeneous blocks in the page can be bounded by rectangular regions, separated by white space or by horizontal and vertical lines. The document is cut alternatively horizontally and vertically according to white spaces based on projection profile of black pixels in the vertical and horizontal direction.

Liu et. al. proposed a hybrid algorithm [LiuT97]. The main idea of this algorithm is performing splits and merges of the regions of the document at the same time. If a region is inhomogeneous, split it into four rectangular sub-regions based on the projection profiles. If two adjacent regions are homogeneous and their union is also homogeneous, then merge them. Repeat the steps until no splitting and merging can be performed.

The Voronoi based algorithm, which was proposed by Kise et al., is a bottom-up approach [Kise98]. This algorithm extracts points on the boundaries of the connected components and draws the voronoi cells surrounding those points. Finally, a large number of superfluous Voronoi edges (which are like the edges lying between characters, words and text lines) are removed.

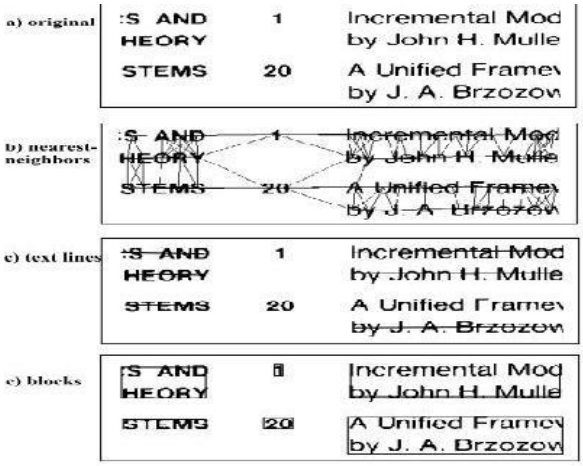


Figure 4: Gorman's method a) Original portion of table of contents page. b) Nearest neighbors. c) Find text lines. d) Structural blocks [Gorm93].

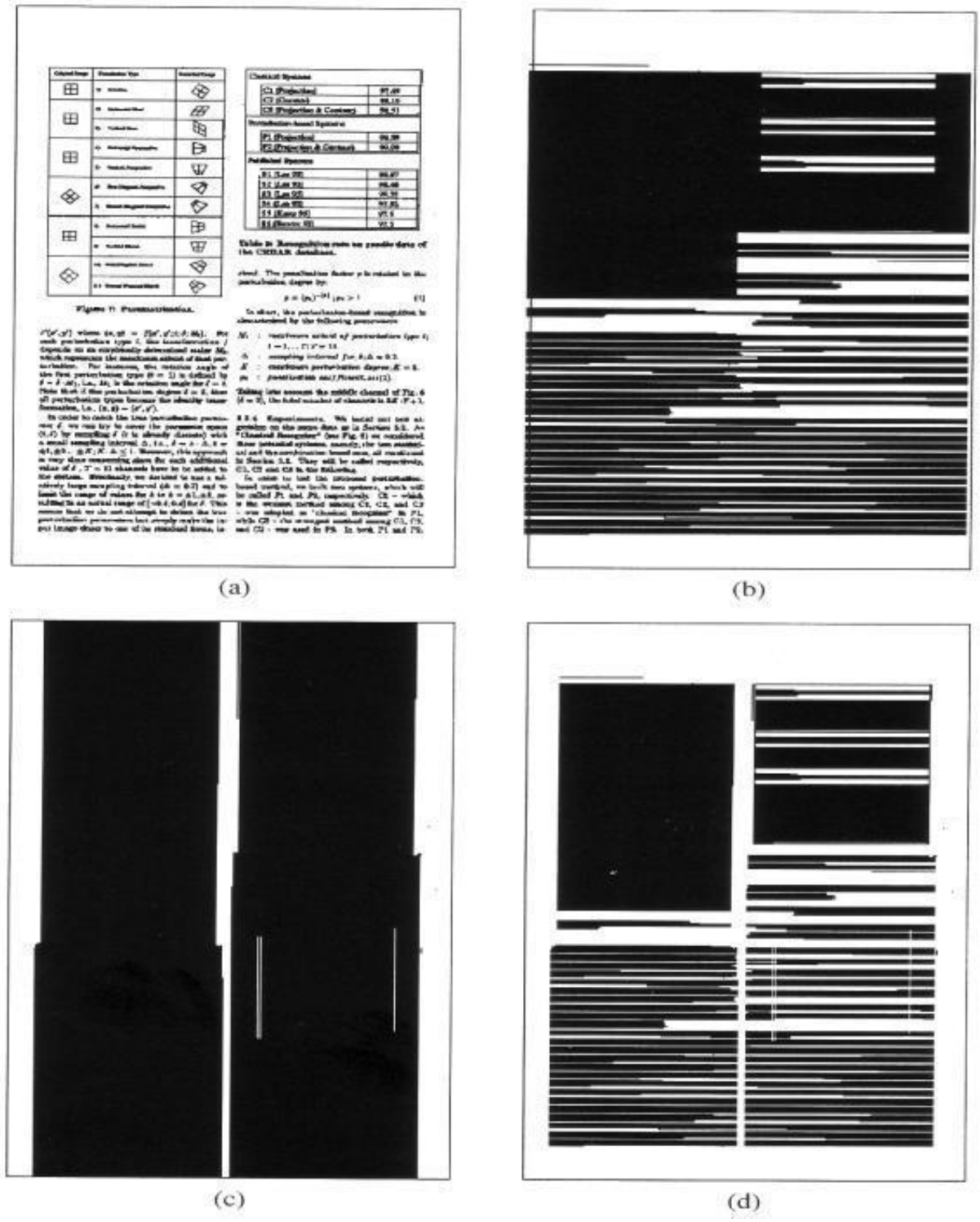


Figure 5: The smearing method: a) Original page. b) Smearing horizontally. c) Smearing vertically. d) Combined (b) & (c) by a logical "and" operator [Wong82].

Hadjar and Ingold presented an algorithm to segment Arabic documents into homogenous regions [Hadj03]. This algorithm is a bottom-up approach based on connected components. All text lines are extracted by : 1) Appling RLSA horizontally, such that if the horizontal distance between any two black adjacent pixels are less than a threshold, those pixels are merged by changing each white pixel between those pixels to black, 2) extracting the connected components as lines. Because Arabic text may contain diacritic points above or under characters, the previous merge process will be applied again with two different thresholds for the diacritics either above or under the characters. Those lines are then grouped into blocks if the distance between the lines is less than another threshold.

An enhanced background thinning method was presented for physical layout in [ChiW03]. The method consists of three phases, namely background thinning, removal of unnecessary chains, and removing chains generated inside a halftone region during thinning. In the background thinning algorithm, edge points of a region are recursively deleted if the deletion of these points does not remove end points, does not break connectedness, and does not cause excessive erosion of the region. Then the unnecessary chains, which lie between text lines and between characters, are removed if the chain satisfies the following criterion:

$$(T_D * W + T_W * D) - (T_D * T_W) \leq 0$$

where D is the minimum city block distance to black pixel, W is the difference of the average line widths, and T_D and T_W are the pre-set thresholds of D and W respectively.

In the last phase after tracing the boundary of each region, the unnecessary chains generated inside a halftone region boundary during thinning are removed to generate the final result of the page segmentation method.

A Persian/Arabic document segmentation method based on the spreading of ink on paper was presented in [Shir05]. The steps of this method , which are shown in Figure 6, are preprocessing, simulating the ink spread effect, and identifying the parts of the image. In the preprocessing, small connected components less than a threshold are identified as noise and big components (which are greater than a threshold) are identified as non text. Noise and non text components are removed. In simulating the ink spread effect step, a spread radius was defined in equations (2.3), and (2.4) for smaller or bigger fonts respectively. Each pixel that has a white neighbor (a circle), whose center is this pixel and radius is the spreading radius, is drawn in black. In identifying the parts of the image step after spreading, each of the connected components identify a certain part of the main image as regions. At last, the connected components of the resulting image are used as a mask for the image obtained from the preprocessing step to produce the final output (page layout).

$$Radius_1 = (\text{Num. of black pixels in the component})/(20) \quad (2.3)$$

$$Radius_2 = ((\text{Num. of black pixels in the component})/(20))*(0.4) \quad (2.4)$$

(a)

(b)

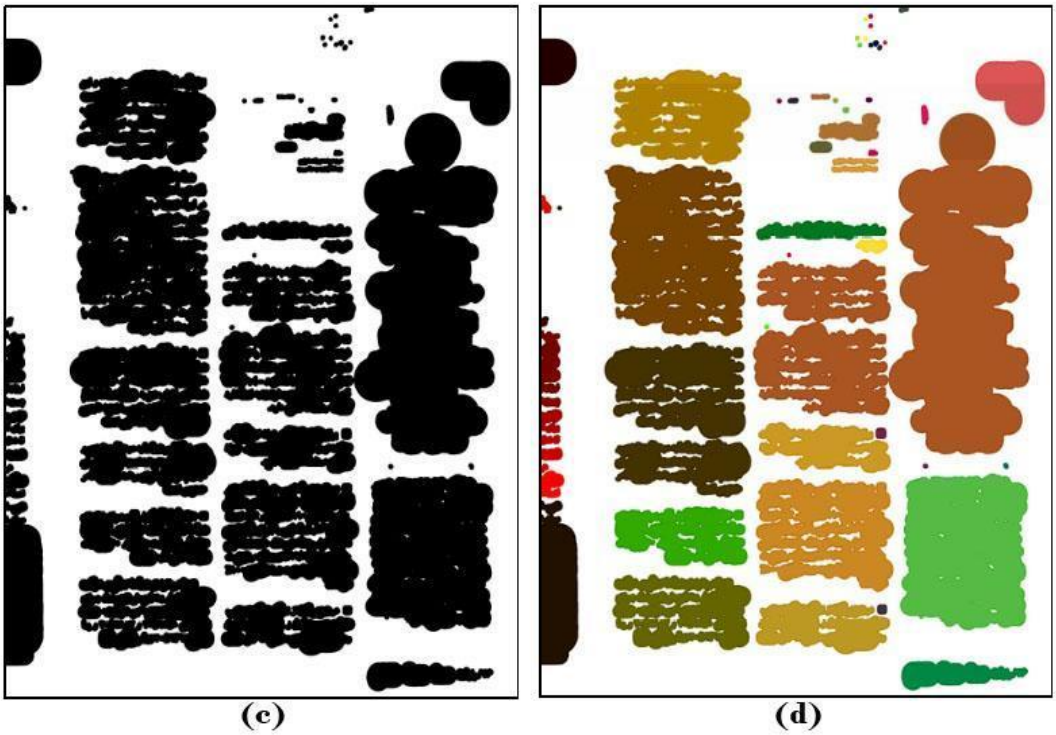


Figure 6: spreading of ink method a) Original page. b) After preprocessing. c) The result of simulating the ink spread effect. d) Mask generated for making output. e) Final Result **[Shir05].**

The survey of Antonacopoulos et. al. is the 5th edition of the ICDAR page segmentation competition series [Anto09]. This competition was held in the context of ICDAR2009. The survey described four methods whose results had been submitted to the competition. The first method is DICE system (Document Image Content Extraction) which is based on pixel classification. The DICE system consists of two main steps: per-pixel [Bair07] and post-classification methodology [AnBa07]. In the per-pixel classification, individual pixels were classified mainly into machine-print text, handwriting text or photograph. In the post-classification methodology, classes are reassigned to favor local uniformity. Each pixel sample was represented by scalar features extracted from a small region (e.g. 4-5 pixels radius) centered on that pixel.

The second method is the Fraunhofer Newspaper Segmenter (Antonacopoulos et. al. 2009). This method included five phases. The first phase is pre-processing that applies global optimal binarisation to the input grey-scale image. Next phase is black separator detection to detect horizontal and vertical lines. The third phase is white separator detection to detect maximally empty rectangles on some conditions that their height must be large enough in relation to the dominant character size. The fourth phase is page segmentation which is a hybrid approach that is a bottom-up process guided by top-down information given in the form of logical column layout of the page. The final phase is text line and region extraction which computes font characteristics (e.g. stroke width, x-height, italics) for each text line to extract the text regions with similar properties.

The REGIM-ENIS is the third method which is designed for degraded multi-script multi-lingual complex official documents (Antonacopoulos et. al. 2009). This system consists of five classes: tabular structures, logos, stamps, handwritten text and images. It includes two stages. In the first, the page is segmented into text and non-text regions based on a steerable pyramid transform in the feature extraction phase. Script identification to the printed and handwritten regions is performed in the next stage.

The last system is the Tesseract method which includes binary morphology and connected component analysis to estimate the type (text, image, separator, or unknown) of the connected components (Antonacopoulos et. al. 2009). The regions of a page are bounded by tab-stops detection to mark out column boundaries, indents, table columns etc. An analogous process is applied to strips of image regions, so that they may also be wrapped into a poly-rectangle shape where text is flowed around

non-rectangular images. The column layout also defines the physical reading order for the detected regions.

2.2.2. Document Contents' Classification

To label each region of the documents as text, drawings etc. we need to extract features for those regions. For this reason, this section reviews some related works which addresses features extraction and classification.

In [Ingl95], some features were presented to classify each zone into text, halftone pictures, line drawings, or mathematical Formulae. The used features are the number of components per unit area (number of foreground pixels), aspect ratio, circularity, and separation. The aspect ratio feature is the mean ratio of height to width of the components' bounding boxes. The circularity is the square of the perimeter of a zone divided by the average of connected components' area. The perimeter of the zone is as given by the following equation:

$$Perimeter = 2 * H + 2 * W$$

Where H and W are the height and the width of the zone respectively. In the separation, the distance between each connected component in the zone and its nearest component is computed. Then the average of the distances is taken.

In [Sauv95], document image was divided into small n*n pixel windows. Then a number of features were extracted from each windows. Those features were black/white-ratio inside the single window, average horizontal black run-length and vertical cross-correlation between neighbouring pixels and between the first and every fifth (relative) pixel. Rule-based classification is used to classify the page contents into background, text, and picture.

Density features and connectivity histogram are extracted in [FanW95] to classify each zone into text, graphics, and image. First, the density feature was utilized to determine whether the zone is a text or non-text. The connectivity histogram is employed to classify non text zones into graphics or image block. The database was 30 English and Chinese documents. The reported average classification accuracy is 94%.

Liang et. al described a feature based supervised zone classifier using the information of the widths and heights of connected-components within a given zone [Lian96]. First of all, the bounding boxes of each connected component are selected. The height and width of the connected components are defined as the height and width of their bounding box. Then the histogram of the connected components' bounding boxes' widths and heights for each zone is computed. They observed that, each different zone usually has different distributions of connected component heights and widths. For example, a text zone has many connected components with the size of individual characters. The distribution of the widths and heights is encoded into n*m dimensional feature vector, where n and m are the number of intervals for the widths and heights respectively. The feature values are the normalized connected components' numbers corresponding to the different height and width intervals. As the authors' experiment, the values of n and m are 10 and 11 respectively. This system used a binary decision tree to classify each zone into one of eight classes (viz. text of font size 8-12, text of size 13-18, text of size 19-36, math, table, line drawing, halftone, and ruling). The used data set is UW-I database. They reported accuracy for text and non-text distinction greater than 97%.

Wang et. al. presented a zone (block) classification system [Wang06]. This system used optimized decision tree and Hidden Markov Model (HMM) classifiers to classify each block to one of nine classes, 2 text classes (font size: 4 − 18pt and 19 − 32pt), math, table, halftone, map/drawing, ruling, logo, and others. HMM was applied on some set of zones located on headers or footers of document images. The data set is the University of Washington III database (UW-III) which consists of 1600 images with a total of 24177 zones. The performance, as the authors claimed, was 98.45%. Twenty five features were extracted from each block. These features are such as run length, spatial, background, etc. A run length feature is the average and the variance of pixels in a given foreground or background run in each of the four directions (viz. horizontal, vertical, left-diagonal, and right-diagonal). The spatial features are the spatial mean and the spatial variances in each of the four directions, horizontal, vertical, left-diagonal, and right-diagonal. Equations (2.5) and (2.6) describe respectively the spatial mean ($\mathrm{spmean_p}$) and the spatial variances ($\mathrm{spvar_p}$) in the four direction.

$$\mathrm{spmean_p} = \frac{1}{f} \sum_{l \in L_p} w_p \times \mathrm{proj_{p,l}} \tag{2.5}$$

$$\mathrm{spvar_p} = \frac{1}{f} \sum_{l \in L_p} \left[\mathrm{proj_{p,l}} \times \left(w_p - \mathrm{spmean_p} \right)^2 \right] \tag{2.6}$$

Where f is the foreground pixel set in a given zone, p represents the four directions (horizontal (h) , vertical (v), left-diagonal (l), and right-diagonal (r)), $\mathrm{proj_{p,l}}$ is the sum of run lengths in a given direction, and L_p is the set of projections in the four directions. For each projection, there are start (x_s, y_s) and end (x_e, y_e) points; w_p is a weight whose value depends on the direction. For the horizontal direction, w_p equals

to y_s. In the vertical direction, w_p equals to x_s. In the left-diagonal direction, w_p equals to $x_s + y_s$. In the right-diagonal direction, w_p equals to $x_s + y_s$.

The background features are categorized into two types of features. The first type is the ratio of the number of background pixels to the total number of the pixels in the zone. The second type is the total area of the large horizontal and large vertical blank blocks. Where any horizontal blank block is a large horizontal blank block if it does not touch the left or the right sides of the zone bounding box and its column numbers are larger than a threshold. A vertical blank block is a large vertical blank block if it does not touch the upper or the bottom sides of the zone bounding box and its row numbers are larger than a threshold.

Keysers et. Al. provided a document block classification system based on run-length histogram feature vector alone [Keys07]. This system used a novel combination of known algorithms to combine different features which were collected from other works. Those features were collected from [Dese04], [Wang06], and [Okun99]. They classified blocks to one of eight classes math, logo, text, table, drawing, halftone, ruling, and speckles. They used k-nearest neighbor classifier. The classifier used UW-III data set consisting of 1600 images with a total of 24177 zones. They reported an error rate of 1.46%.

Bloomberg presented an algorithm, which uses multiresolution morphology based threshold reduction operation, to separate text and halftone image from a document image [Bloo91]. The main idea of the threshold reduction operation is that when the image is tiled into 2 * 2 pixel blocks, each 2*2 block of four pixels is replaced by '1' if the summation of the fours pixels are greater than or equal to a threshold T,

otherwise ‘0’. This algorithm applies the threshold reduction operation twice with threshold value of 1, then it uses the operation twice with difference threshold values, which are 4, and 3. Bukhari et. el. presented an improvement of this algorithm and used it to segment Text and Non-Text components in Arabic and Urdu Document Images [Bukh11]. The steps of their algorithm are applying the threshold reduction operation twice with threshold values of 1, reconstruction of broken drawing lines, a hole-filling morphological operation applying the threshold reduction operation with a threshold of 4, and applying the threshold reduction operation with a threshold of 3. Reconstruction of broken drawing lines step is employed to connect any broken drawing lines. The main idea of the hole-filling morphological operation is filling the hollow contours before the threshold reduction operations with high thresholds because the threshold reduction operations with higher thresholds remove these hollow contours. The reason of revealing the hollow contours is because non-text components, such as drawings, maps, graphs and even halftones, are often composed of hollow contours of geometric and irregular shapes. Figure 7(b) shows the non-text components extracted.

بمناسبة الاحتفال بعيدها الـ٦٥

«زيروكس» تطلق أجهزة وحلولاً جديدة في «جايتكس»

تعتزم شركة زيروكس العالمية العاملة في مجال أنظمة إدارة الوثائق إطلاق تشكيلة جديدة من أحدث ابتكاراتها في عمليات الطباعة والأجهزة المتعددة الوظائف وذلك ضمن فعاليات معرض جيتكس في دبي لمناسبة مرور ٦٥ عاماً على تأسيسها.

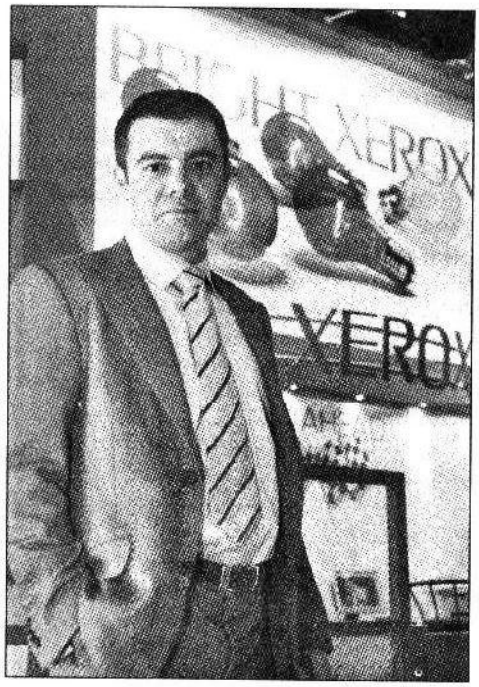
وعلق بن غيل، المدير العام، قسم المبيعات والتسويق، زيروكس الشرق الأوسط وافريقيا، قائلاً: «منذ بداياته الأولى، كان معرض جيتكس، ولا يزال، يشكل فرصة ذهبية بالنسبة لشركة زيروكس لكي تعرض منتجاتها المبتكرة والحديثة على جمهور مناطق الشرق الأوسط وافريقيا. كما أنه يشكل أيضاً الملتقى المثالي الذي تستطيع زيروكس من خلاله الالتقاء والتفاعل مع شركائها الأعزاء وعملائها الكرام الذين يتوافدون الى المعرض من جميع أرجاء المنطقة».

News From Xerox

كما ستنتهز زيروكس الشرق الأوسط وافريقيا فرصة المعرض لكي تعلن عن اتفاقية شراكة إقليمية رئيسية جديدة في إطار سعيها المتواصل الى تعزيز شبكتها التوزيعية في المنطقة.

ونجحت زيروكس في تعزيز مستويات الانتاجية والأمان التي تتمتع بها أجهزتها متعددة الوظائف بواسطة تطوير منصة برمجية جديدة قادرة على تلبية الاحتياجات المختلفة لقطاعات الفندقية وتجارة التجزئة والشركات المالية والرعاية الصحية. إن واجهة زيروكس القابلة للبرمجة، Xerox Extensible Interface Platform، تفتح آفاقاً واسعة من الامكانيات أمام أجهزة زيروكس متعددة الوظائف. فهي تتيح للمستخدمين تعديل الجهاز لكي يلائم متطلباتهم الخاصة تمام الملاءمة، بدلاً من أن تضطر المؤسسة الى التكيف مع مواصفات الجهاز العامة.

سوف يتواجد في منصة زيروكس السيد أندرز لوسادا، مدير التسويق في أنظمة زيروكس، لكي يمنح زوار المنصة فكرة شاملة حول التشكيلة الواسعة من الأنظمة التي يمكن دمجها مع أجهزة زيروكس متعددة الوظائف لتقديم مزايا حقيقية وفوائد لا تضاهى الى العملاء.



● مدير عام «زيروكس» بن غيل

كما ستسلط زيروكس الأضواء على الأجهزة متعددة الوظائف ذات الأسعار الاقتصادية، والمصممة خصيصاً للاستخدام في الشركات والمكاتب الصغيرة والاستخدام المنزلي، حيث تقدم مستويات رفيعة من الأداء والجودة والسرعة بأسعار اقتصادية لا تنافس. وسوف يقوم فريق عمل زيروكس المتواجد في المعرض بعرض مزايا أنظمه زيروكس الملونة على الزوار من الأفراد وأصحاب الشركات الصغيرة، ويشرحوا لهم كيف يمكن لكلفة الامتلاك الكلية المنخفضة أن تؤدي الى تحقيق عوائد استثمارية من الصعب جداً منافستها.

(a)

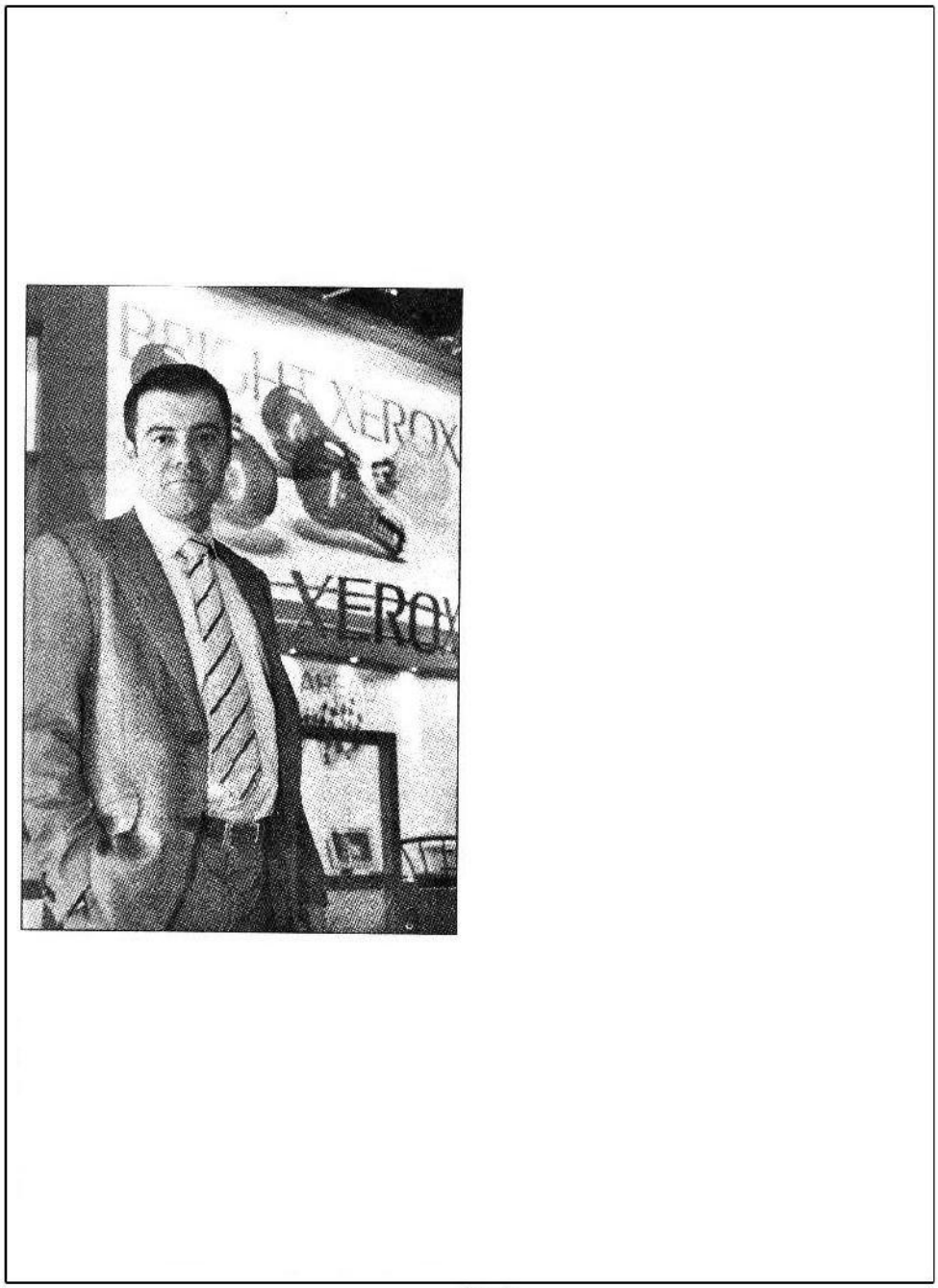


Figure 7: Bukkari et al.'s method: (a) the original image (Arabic document); (b) the non-text components extracted from (a) [Bukh11].

2.3. Script Identification

When a text region has multilingual scripts, we need to group these scripts in order to apply each group to its suitable OCR. Existing script identification approaches can be categorized into two types, namely, local and global approaches.

In local approaches, features are extracted from a documents at the line, word or character levels. While in global approaches, the features is extracted from document at the page, or text block levels.

In [Hoch97], every text symbol (character, word, or part of a word) in a document is matched to a set of template symbols to distinguish thirteen scripts (Arabic, Armenian, Brumese, Chinese, Cyrillic, Devnagari, Ethiopic, Greek, Hebrew, Japanese, Korean, Latin and Thai). For training set, the connected components (symbols), whose area is less than 10 pixels or bigger than 80 pixels are eliminated in order to eliminate flecks and borders. To generate the templates for each script, the other symbols are rescaled to 30 * 30 pixels using a standard nearest-neighbor resampling algorithm. Then similar symbols within each script are clustered as templates for this script. Based on Hamming distance, every textual symbol in an input document is matched to a set of template symbols and is classified to the script class of the best matching template symbol.

In [Spit97], the presence and the location of upward concavity is used as a feature to separate Latin and Hangul (Japanese, Chinese, and Korean languages) scripts. By examining sets of runs within the connected component, the presence and location of upward concavities are determined if two runs of black pixels appear on a single scan line of the raster image.

In the category of local approaches, Elgammal and Ismail presented a method to separate Arabic and Latin text in both printed and handwritten scripts [Elga01]. This method

addresses the script identification problem at the word and text line levels. They used a neural network classifier. The features are based on horizontal projection profiles, and run-length Histogram. The features based on horizontal projection profiles detect the peaks and the moments of the horizontal projection profile. The projection profiles for text line and word levels are taken and normalized. Each peak that has height bigger than a certain threshold will be counted and located. Then the number of peaks and their locations in the textline will be selected as features. They choose these features because Arabic text line usually have one peak around the middle of the line while Latin text line has two peaks, one in the upper half of the line and one in the lower half. Figure 8 shows a sample of a typical normalized profile for Arabic text line. The moments of the horizontal projection profile are computed according to the following equations:

$$m_r = \frac{\sum_{y=y_{bottom}}^{y_{top}} (f(y) - \bar{h})^r}{y_{bottom} - y_{top} + 1}$$

$$\bar{h} = \frac{\sum_{y=y_{bottom}}^{y_{top}} f(y)}{y_{bottom} - y_{top} + 1}$$

Where m_r is the r^{th} moment, $f(y)$ is the height of the horizontal projection profile at row y, $y_{bottom} \leq y \leq y_{top}$, and $\bar{h}$ is the mean height of the horizontal projection profile at all rows between y_{bottom} , and y_{top}. The values of the third, fourth and fifth moments (m_3, m_4, m_5) for a group of Arabic and English text lines with different lengths are taken as the features. For run-length Histogram feature, the text line is divided into 8*8 bins and the histogram of the run-lengths in each bin is calculated. Their reported performance is 99.7% for text line level and 96.8% for word level.

Figure 8: The normalized projection profiles [Elga01]: a typical The normalized projection profiles The normalized projection profiles profile for Arabic text line. Where y_{bottom} and y_{top} are the top and bottom of the text line.

Kanoun et. al. proposed in [Kano02] a script identification method to differentiate Arabic and Latin homogeneous text blocks for printed and handwritten scripts. This method uses morphological and geometrical analysis. Five morphological features are extracted at the text block level and 16 geometrical features are extracted at the text line and the connected components levels. Their reported highest correct identification rate is about 96.1 % when they combined some features, namely bottom and top diacritics percentage, bottom occlusions percentage, average connected components eccentricity, average and standard deviation of black pixel density of connected components, average occlusions' eccentricity, standard deviation of contour pixels density of connected components, and average of connected components' spheroid (or average of contour pixels density of occlusions or standard deviation of text lines height). This method used K- nearest neighbors with K = 5 and the Hamming distance.

In [Busc05], Busch et al. evaluated number of texture features including gray-level co-occurrence matrix, Gabor filter bank energies, and a number of wavelet transform-based features extracted at the text block level. The height and width of text blocks are normalized by standardizing the spaces between the lines and the words.

Global approaches are designed for analyzing text blocks extracted from the document image. In [Jlai07], three decision levels' strategy is proposed to differentiate between Arabic and Latin script in printed and handwritten text. The first level classifies the text block into two classes which are printed Latin, and other (i.e. printed and handwritten Arabic and handwritten Latin). The features of the first level are the average and standard deviation of following: the eccentricity of each connected component (the width /the height), the density of black pixels of each connected component, and the length of each separator between two connected components. The second level is used to separate two classes. The first class is the printed Arabic while the second includes handwritten Arabic and handwritten Latin in the same class. The features of this level are the average and standard deviation of following: the eccentricity of each occlusion of the text block, density of internal contour pixels of each occlusion. The last level distinguishes handwritten Arabic and handwritten Latin. The feature is the total number of the diacritic points at the bottom of the baseline divided by the total number of the diacritic points at the top. K- nearest neighbors with K equals to 5 and Hamming distance were used. Their reported performance is 95%.

CHAPTER 3

DOCUMENT ANALYSIS AND CLASSIFICATION

The document analysis and classification system has four components: preprocessing, document segmentation, text feature extraction, and classification. A document image is enhanced in the preprocessing phase by removing noise, binarization, and detecting and correcting image skew. The image is segmented into homogeneous regions in the document segmentation phase. Then each region is classified into text or non text in the classification phase based on a number of features that are extracted from each region in the feature extraction phase.

In some literature works, regions are classified into text, math, table, halftone, map/drawing, ruling, logo, etc. In our research, we represent the text and math classes as text classes, while we represent the other classes as non text classes.

3.1. Preprocessing

Preprocessing is necessary to improve document images before the segmentation phase. The preprocessing includes binarization, noise removal, and skew detection and correction.

The Otsu algorithm [Otsu79] is used to identify the threshold, used to convert a gray-level image into binary image. So, each pixel on the image is transformed into white (foreground) if its gray level is smaller than the threshold, otherwise it set to black (background).

For noise removal, the Statistical Based Smoothing algorithm [Mahm94] is employed.

3.2. Document Segmentation

3.2.1. Introduction

In document segmentation, a document is divided into homogeneous regions, then each region is classified into text or non-text region based on a number of features. We proposed an algorithm to partition a document into homogeneous regions and extract a number of features from those regions. To evaluate and compare the proposed algorithm with existing segmentation algorithms, the run-length smearing algorithm (RLSA) [Wong82], and X-Y cut [Jaek95], [Nagy92] are implemented.

3.2.2. Proposed Segmentation Algorithm

The proposed segmentation algorithm, which is shown in Figure 9, consists of three steps. In the first step, the image is divided into a number of n*n pixel windows. Each n*n pixel window is replaced by '0' if the number of the black (foreground) pixels in the window are more than a threshold otherwise '1'. This task, shown on Figure 10, removes some noise and rescales the image into a smaller size. The boundaries of each connected components of the scaled image are found. If any black pixel is not located in the boundaries, it will be ignored. Finally, the connected components of the resulting image are assigned as regions excluding internal components. Figure 11 shows some samples, which are fed to the algorithm.

The Proposed Segmentation Algorithm:

1 Scan the image by an n*n window. Create a binary matrix 'C' depending in the following rules (where each cell represents the n*n window in the image as shown in Figure 10).

   a. If all the pixels of the window are black, assign 0 to the corresponding cell of the matrix
   b. If all the pixels of the window are white, assign 1 to the corresponding cell of the matrix
   c. If the window has mixed (white and black) pixels, then
      i. If the black pixels in the window are less than Threshold, assign 1 to the corresponding cell of the matrix
      ii. Otherwise assign 0.

   The matrix 'C' represents the scaled image, where '0' cells represent black (foreground) pixels in the scaled image and '1' cells represent white (background) pixels.

2 Find boundaries of the connected components in the scaled image. Change each black pixels into white pixels except the border pixels and store the result in a new image called "perimeters". Figure 11.(c) shows the result of this step.

3 Each connected component on "perimeters" is assigned as a region if it is not internal component. If a connected component is an internal component, merge it with its container component.

Figure 9: The Proposed Segmentation Algorithm

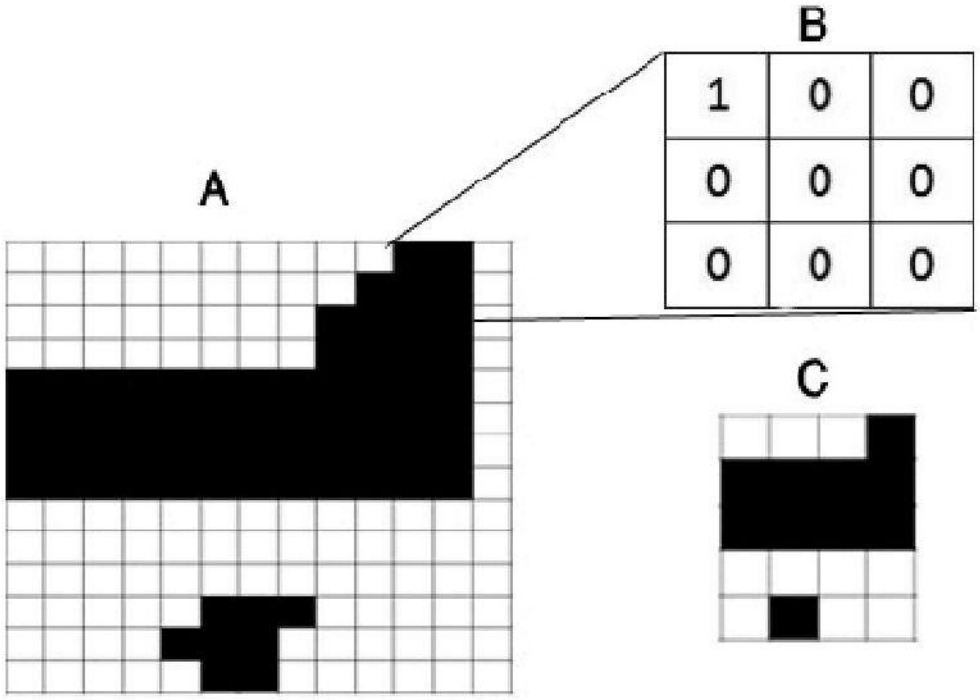


Figure 10: an example of rescaling the images with 3*3 window.

السيطرة على حريق مزارع «خويلدية القطيف»

القطيف – ماجد الشبركة

تمكنت فرق الدفاع المدني بالقطيف من محاصرة وإخماد حريق شب ظهر أمس، في إحدى مزارع بلدة الخويلدية التابعة للمحافظة. واندلع حريق ظهر أمس، في إحدى المزارع الواقعة غرب بلدة الخويلدية والقريبة من أحد الأحياء السكنية، ونسبب الحادث الذي لم تعرف أسبابه بعد، في احتراق جلسة وغرفة صفيح وعدد من أشجار النخيل وإحداث سحابة كثيفة من الدخان، وتطاير بعض الشرر إلى الأماكن المجاورة للمزرعة، فيما ساهمت سرعة وصول فرق الدفاع المدني للموقع ومحاصرتها لألسنة اللهب وإخمادها للحريق فورا في الحد من انتقاله للمزارع المجاورة.

ألسنة اللهب والدخان تتصاعد من موقع الحريق (الشرق)

(a)

السيطرة على حريق مزارع «خويلدية القطيف»

القطيف – ماجد الشبرقة

تمكن فرق الدفاع المدني بالقطيف من محاصرة وإخماد حريق شب ظهر أمس في إحدى مزارع بلدة الخويلدية التابعة للمحافظة. وإندلع حريق ظهر أمس في إحدى المزارع الواقعة غرب بلدة الخويلدية والقريبة من أحد الأحياء السكنية، وتسبب الحادث الذي لم تعرف أسبابه بعد في احتراق جلسة و غرفة صفيح وعدد من أشجار النخيل وإحداث سحابة كثيفة من الدخان وتطاير بعض الشرر إلى الأماكن المجاورة للمزرعة، فيما ساعدت سرعة وصول فرق الدفاع المدني للموقع ومحاصرتها لألسنة اللهب وإخمادها الحريق قبل أن يمتد من المزارع المجاورة.

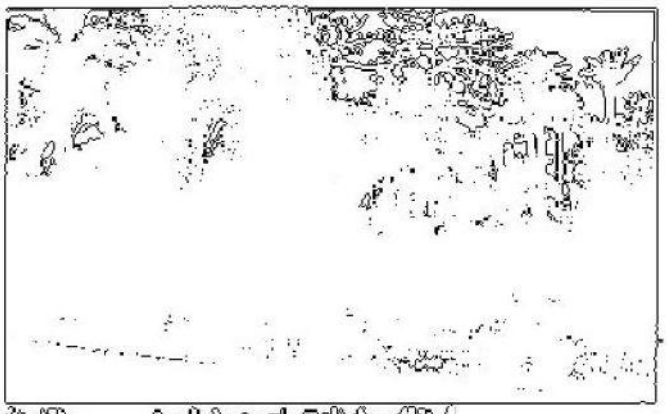
ألسنة اللهب والدخان تتصاعد من موقع الحريق (الشرق)

(b)

السيطرة على حريق مزارع «خويلدية القطيف»

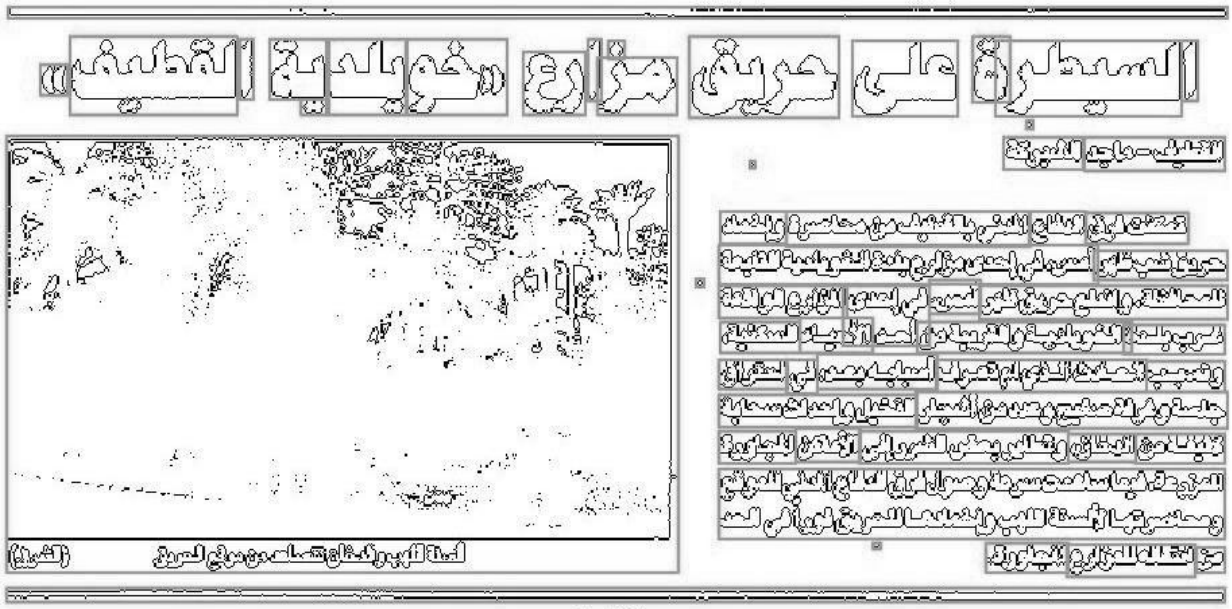
القطيف – ماجد الشبرقة

تمكن فرق الدفاع المدني بالقطيف من محاصرة وإخماد حريق شب ظهر أمس في إحدى مزارع بلدة الخويلدية التابعة للمحافظة. وإندلع حريق ظهر أمس في إحدى المزارع الواقعة غرب بلدة الخويلدية والقريبة من أحد الأحياء السكنية، وتسبب الحادث الذي لم تعرف أسبابه بعد في احتراق جلسة و غرفة صفيح وعدد من أشجار النخيل وإحداث سحابة كثيفة من الدخان وتطاير بعض الشرر إلى الأماكن المجاورة للمزرعة، فيما ساعدت سرعة وصول فرق الدفاع المدني للموقع ومحاصرتها لألسنة اللهب وإخمادها الحريق قبل أن يمتد من المزارع المجاورة.



ألسنة اللهب والدخان تتصاعد من موقع الحريق (الشرق)

(c)



(d)

Figure 11: the proposed algorithm for segmentation; (a) a sample image; (b), and (c)are the result images after applying the first and second steps, and (d) shows the boundaries of the resulting regions

3.2.2.1. Feature Extraction

The following features are extracted from each extracted region.

1. The Foreground / Background Means

   The Foreground/ background Means feature is the ratio of the foreground/ background pixels to the total number of pixels in each region.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (3.1)$$

   Where n are the number of elements. If foreground pixel represents '1', $\bar{x}$ is the mean of the foreground pixels otherwise it is the mean of the background pixels.

The text regions, which have multiple text lines and multiple words, has fairly comparable number of foreground or background pixels. While in non-text region, the number of foreground pixels are always more than the number of background pixels or vice versa.

2. Standard deviation of the foreground pixels

   We assume the foreground pixels represent '1' while the background pixels represent '0'. The standard deviation of each column of the region is computed. Then the mean of the standard deviation vector is calculated as a feature. Equation 3.2 gives the standard deviation equations.

$$\sigma = \sqrt{\left(\frac{1}{(n-1)} \sum_{i=1}^{n} (x_i - \bar{x})^2\right)} \qquad (3.2)$$

3. The number of connected components

   A connected component is a set of connected foreground pixels. This feature is the number of connected components in the zone divided by the zone's area. Where zone's area is the number of foreground pixels. The feature is taken from [Ingl95].

4. Aspect Ratio

   The ratio of height to width of each connected components is calculated in a zone. The average of the aspect ratio, which is used in [Ingl95], is taken as a feature.

   The height and width of the connected components in some non-text regions have big difference, while they are possible to be close in most text regions.

5. Circularity

The circularity, which is taken from [Ingl95], is the square of the perimeter of the zone divided by the average of the areas of all connected components.

6. The Means of The Horizontal/Vertical Projections

   The horizontal and vertical projections are estimated and their means are selected as features.

7. Background features:

   The background features are based on background pixels. These features, which are taken from [Wang06], follow:

   a. Total area of large horizontal blank blocks.

      A horizontal blank block is a large horizontal blank block if it satisfies the following rules:

      1) Its number columns are large enough compared with the current zone. Specifically

$$\frac{b_c}{Col} > \theta$$

         where b_c is the number of columns of the horizontal blank blocks, Col is the number of columns in the current zone, and θ is 0.1 based on [Wang06]'s experiments.

      2) It does not touch left or right sides of the zone bounding box.

   b. Total area of large vertical blank blocks.

      A vertical blank block is a large vertical blank block if it satisfies the following rules:

      1) Its number of rows are large enough compared with the current zone. Specifically

$$\frac{b_r}{rw} > \theta$$

where b_r is the number of rows of the vertical blank blocks, rw is the number of rows in the current zone, and θ is 0.1.

2) It does not touch the upper or bottom sides of the zone bounding box.

8. Run Length (RL) Features

The run length features [Wang06] include foreground/background run length mean and variance in four directions (viz. the horizontal, vertical, left-diagonal, and right diagonal directions) as shown in Figure 12.

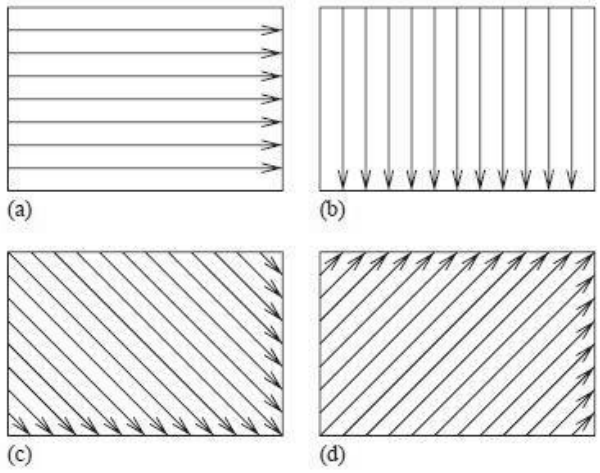


Figure 12: Illustrates the four directions. (a) Horizontal; (b) vertical; (c) left-diagonal; (d) right diagonal.

The run length features are as follows:

1. The foreground horizontal run length mean ($rlmean_h^0$) in a given zone is estimated as follows

$$rlmean_h^0 = \frac{1}{|RL_h^0|} \sum_{rl \in RL_h^0} rl,$$

where RL_h^0 *is the* foreground horrizontal run length.

2. The foreground horizontal run length variance ($rlvar_h^0$) in a given zone is estimated as following:

$$rlvar_h^0 = \frac{1}{|RL_h^0|} \sum_{rl \in RL_h^0} rl^2 - (rlmean_h^0)^2,$$

3. The background horizontal run length mean ($rlmean_h^1$) in a given zone is computed as following:

$$rlmean_h^1 = \frac{1}{|RL_h^1|} \sum_{rl \in RL_h^1} rl,$$

Where RL_h^1 the background horizontal run length.

4. The background horizontal run length variance ($rlvar_h^1$) in a given zone is calculated as follows:

$$rlvar_h^1 = \frac{1}{|RL_h^1|} \sum_{rl \in RL_h^1} rl^2 - (rlmean_h^1)^2,$$

5. The foreground vertical run length mean ($rlmean_v^0$) in a given zone is estimated as follows:

$$rlmean_v^0 = \frac{1}{|RL_v^0|} \sum_{rl \in RL_v^0} rl,$$

Where RL_v^0 is the foreground vertical run length.

6. The foreground vertical run length variance ($rlvar_v^0$) in a given zone is computed as follows:

$$rlvar_v^0 = \frac{1}{|RL_v^0|} \sum_{rl \in RL_v^0} rl^2 - (rlmean_v^0)^2,$$

7. The background vertical run length mean ($rlmean_v^1$) in a given zone is calculated as follows:

$$rlmean_v^1 = \frac{1}{|RL_v^1|} \sum_{rl \in RL_v^1} rl,$$

Where RL_v^1 is the background vertical run length.

8. The background vertical run length variance ($rlvar_v^1$) in a given zone is estimated as following:

$$rlvar_v^1 = \frac{1}{|RL_v^1|} \sum_{rl \in RL_v^1} rl^2 - (rlmean_v^1)^2,$$

9. The Foreground left-diagonal run length mean ($rlmean_l^0$) in a given zone is estimated as following:

$$rlmean_l^0 = \frac{1}{|RL_l^0|} \sum_{rl \in RL_l^0} rl,$$

Where RL_l^0 is the Foreground left-diagonal run length.

10. The Foreground left-diagonal run length variance ($rlvar_l^0$) in a given zone is computed as following:

$$rlvar_l^0 = \frac{1}{|RL_l^0|} \sum_{rl \in RL_l^0} rl^2 - (rlmean_l^0)^2,$$

11. The background left-diagonal run length mean ($rlmean_l^1$) in a given zone is computed as following:

$$rlmean_l^1 = \frac{1}{|RL_l^1|} \sum_{rl \in RL_l^1} rl,$$

Where RL_l^1 is the background left-diagonal run length

12. The background left-diagonal run length variance ($rlvar_l^1$) in a given zone is calculated as following:

$$rlvar_l^1 = \frac{1}{|RL_l^1|} \sum_{rl \in RL_l^1} rl^2 - (rlmean_l^1)^2,$$

13. The foreground right-diagonal run length mean ($rlmean_r^0$) in a given zone is estimated as following:

$$rlmean_r^0 = \frac{1}{|RL_r^0|} \sum_{rl \in RL_r^0} rl,$$

Where RL_r^0 is the foreground right-diagonal run length.

14. The foreground right-diagonal run length variance ($rlvar_r^0$) in a given zone is estimated as following:

$$rlvar_r^0 = \frac{1}{|RL_r^0|} \sum_{rl \in RL_r^0} rl^2 - (rlmean_r^0)^2,$$

15. The background right-diagonal run length mean ($rlmean_r^1$) in a given zone is calculated as following:

$$rlmean_r^1 = \frac{1}{|RL_r^1|} \sum_{rl \in RL_r^1} rl,$$

Where RL_r^1 is the background right-diagonal run length.

16. The background right-diagonal run length variance ($rlvar_r^1$) in a given zone is computed as following:

$$rlvar_r^1 = \frac{1}{|RL_r^1|} \sum_{rl \in RL_r^1} rl^2 - (rlmean_r^1)^2,$$

3.2.3. EXISTING SEGMENTATION METHODS

3.2.3.1. Introduction

We implemented two existing segmentation algorithms, which are the run-length smearing (RLSA) and XY cut algorithms, in order to compare their performance with the proposed segmentation algorithm.

3.2.3.2. Run-Length Smearing Algorithm (RLSA)

RLSA algorithm consists of three steps, viz. horizontal smearing, vertical smearing, and applying logical "and" operator of the vertical and horizontal smeared images. In horizontal smearing, the horizontal white runs, which are smaller or equal to a threshold (horizontal threshold) are changed to black runs. In vertical smearing, the vertical white runs which are smaller or equal to a vertical threshold are changed to black runs. Finally, the result from horizontal and vertical smearing are combined by applying the logical "and" operator. The RLSA algorithm is shown in Figure 13. Figure 14 shows some samples and the results of applying the RLSA algorithm.

The RLSA algorithm:

1. Horizontal_Image, Vertical_Image = input image
2. For HRL = all horizontal white run length on Horizontal_Image
   a. If HRL ≤ Horizontal_Threshold
      i. Change HRL into black.
3. For VRL = all vertical white run length on Vertical_Image
   a. If VRL ≤ Vertical_Threshold
      i. Change VRL into black.
4. Result_Image = (~Horizontal_Image) and (~Vertical_Image)
5. Regions = the rectangle coordinates of the connected components on Result_Image.

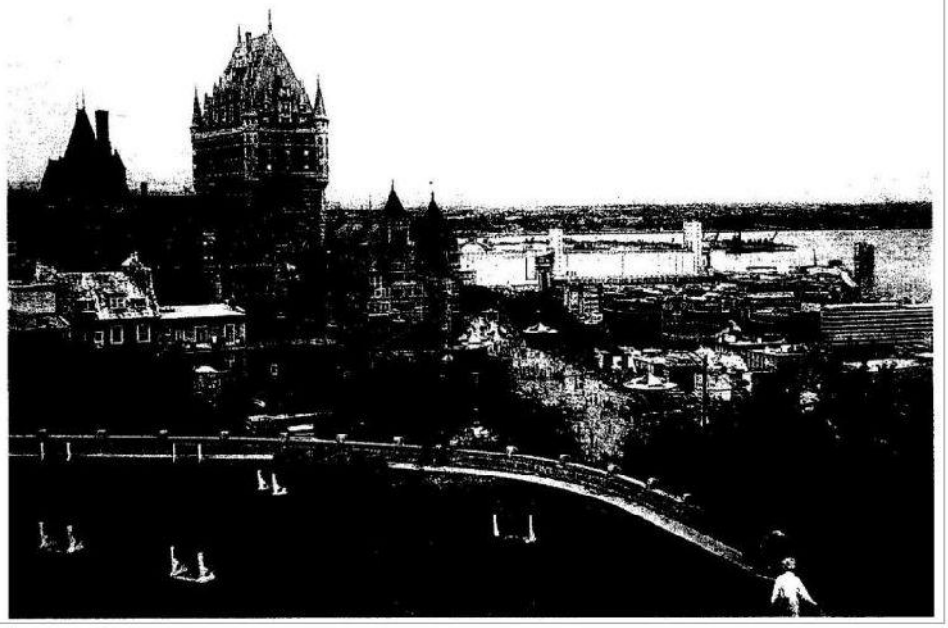
Figure 13: RLSA Algorithm

والفرنسية. أيضًا، تعرض بعض الحكومات الإقليمية تلفزيونهم التربوي العام، مثل TV Ontario في أونتاريو، وتلفزيون كويبيك في الكويبيك. إن موقع كندا المجاور لأكبر منتج مهيمن على صناعة الأفلام الطويلة (هوليود)، جعل صناعة السينما الكندية تحتاج إلى مساعدة كبيرة من الحكومة. فمنذ عام ٢٠٠٠ إلى الآن، تأتي نصف ميزانية الأفلام الكندية العادية من المصادر الحكومية، الإقليمية والاتحادية المختلفة.

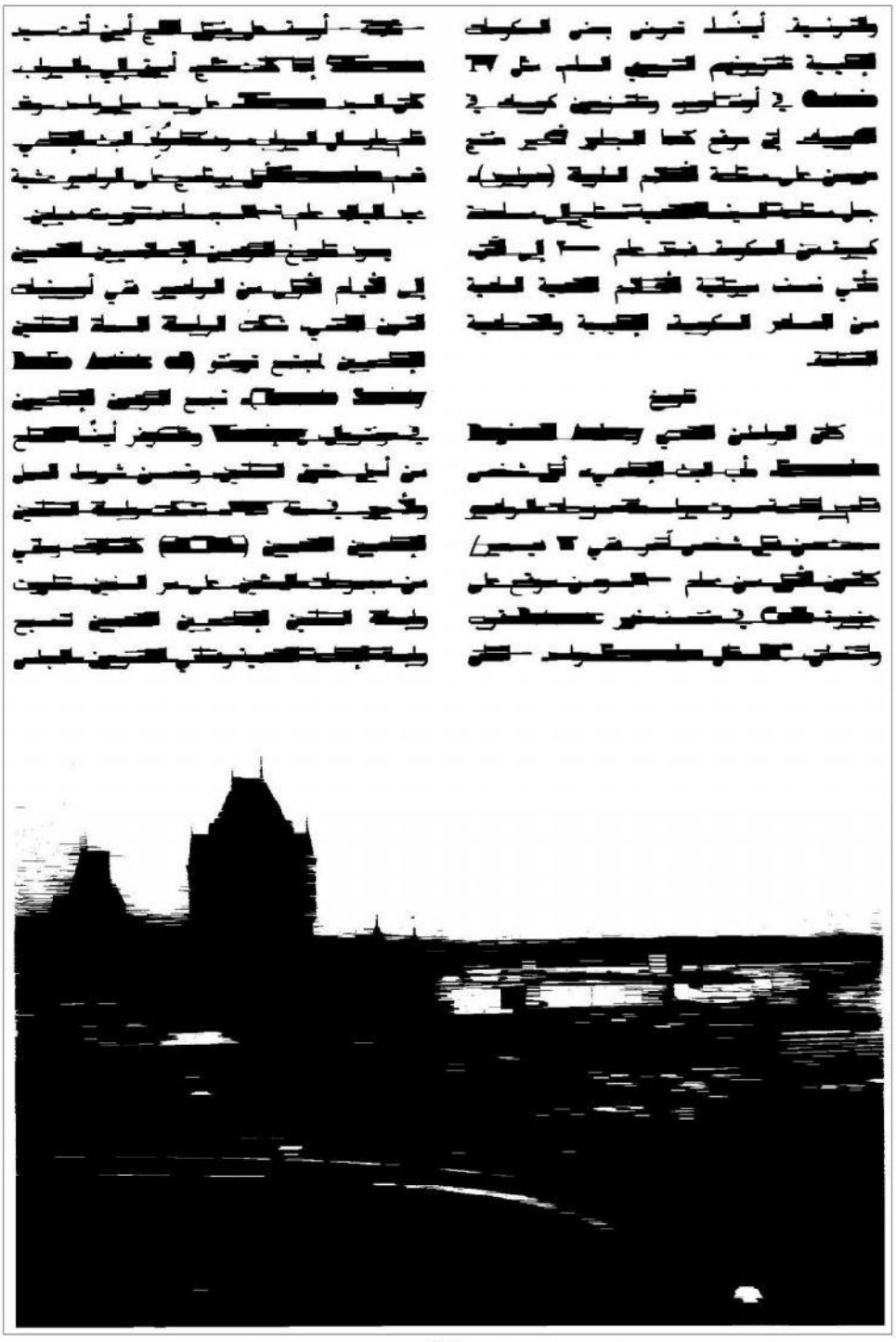
تاريخ

كان المواطن الكندي Reginald Aubrey Fessenden، «أب البث الإذاعي»، أول شخص قام بإذاعة صوت بواسطة موجات الراديو سمعت من قبل شخص آخر. ففي ٢٣ ديسمبر/ كانون الأول من عام ١٩٠٠، ومن موقع على جزيرة Cobb في منتصف نهر Potomac قرب واشنطن، DC، قال Fessenden «واحد – اثنان – ثلاثة – أربعة، هل ينزل الثلج أين أنت سيد Thiessen؟ إذا كانت تثلج، أبرق إلي بالجواب. كان السيد Thiessen، على بعد ميل واحد، وقد قام بارسال الجواب»، وهكذا وُلِدَ البث الإذاعي. قدم Fessenden أول برنامج على الراديو عشية عيد الميلاد عام ١٩٠٦، بمساعدة زوجته هيلين.

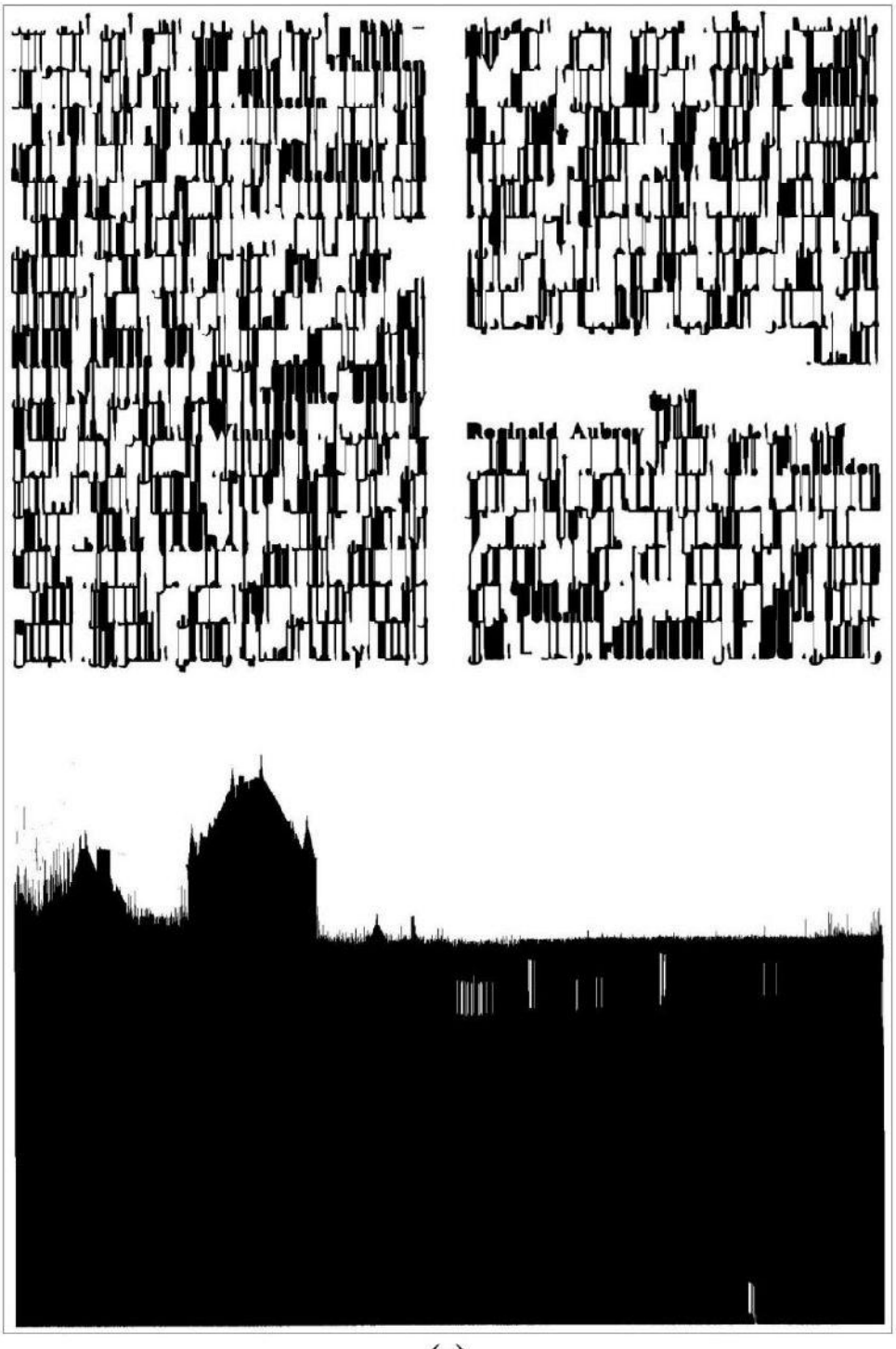
يعود تاريخ المؤدين الإعلاميين الكنديين إلى الأيام الأولى من الراديو. ففي أربعينيات القرن الماضي، شكلت الرابطة المسماة الفنانين الإذاعيين لمجتمع تورنتو (Radio Artists of Toronto Society)، تجمع المؤدون الإذاعيون في مونتريال، Winnipeg وفانكوفر أيضًا للكفاح من أجل حقوق الفنانين، وتحسن شروط العمل والأجور. في سنة ١٩٤٣، تشكلت رابطة الفنانين الإذاعيين الكنديين (ACRA) كائتلاف وطني من مجموعات الممثلين. على مر السنين، تطورت رابطة الفنانين الإذاعيين الكنديين لتصبح رابطة الإذاعة الكندية وفناني التلفزيون، المجلس



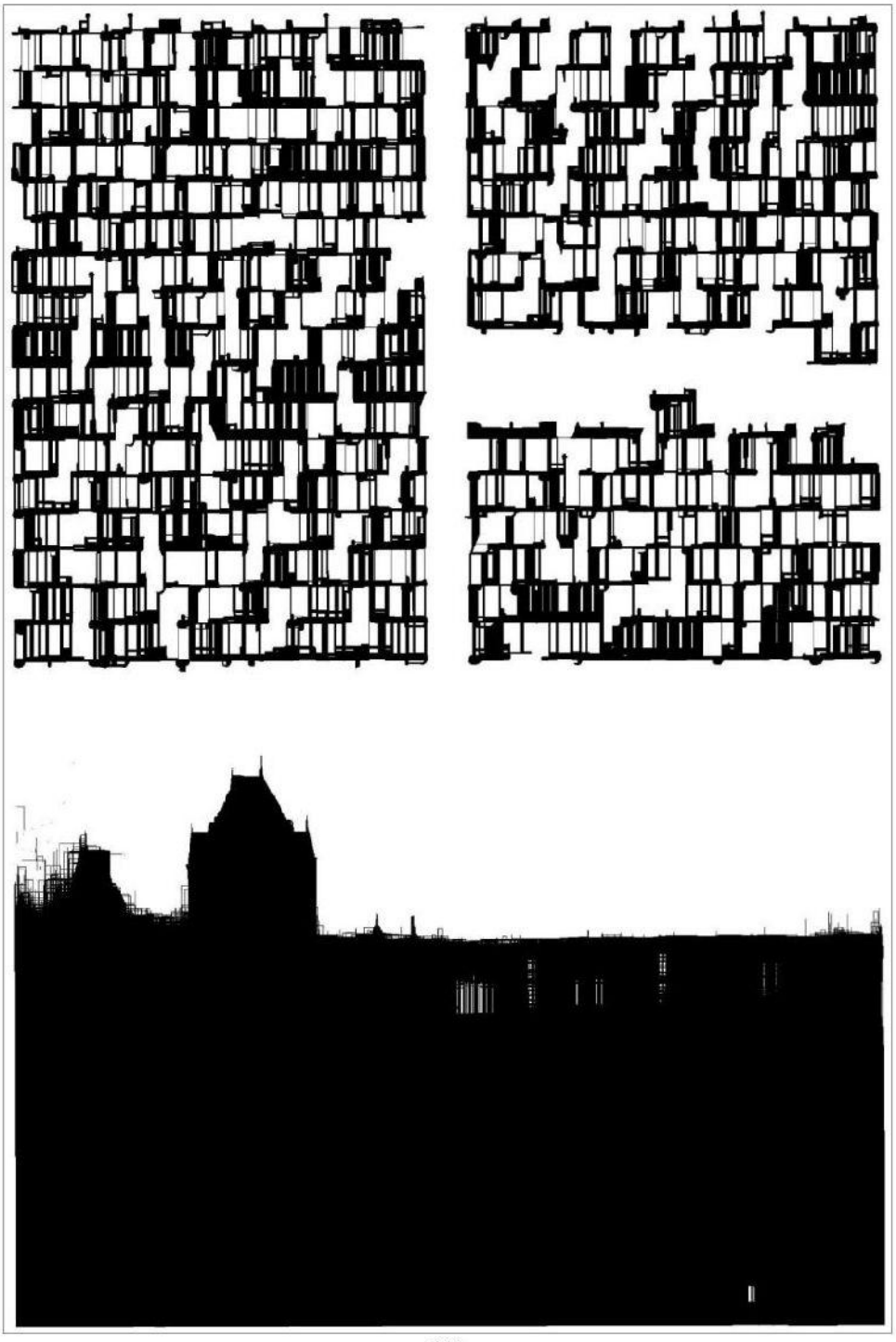
(a)



(b)



(c)



(d)

Figure 14: samples of applying smearing method: (a) Original page. (b) Smearing horizontally. (c) Smearing vertically. (d) logical of b&c

3.2.3.3. XY Cut algorithm

The XY cut technique attempts to repeatedly subdivide an image if it has white cuts whose width or height are bigger than thresholds. So, a page is subdivided repeatedly into smaller rectangles by making cuts along horizontal and vertical directions until it cannot further partition the rectangle. Figure 15 shows the XY cut algorithm that includes two algorithms; the main and the secondary algorithms. Figure 16 shows some samples and the results of applying XY cut algorithm

The XY cut main algorithm:

The Input Parameters: the image, the indexes of the first and last rows, and first and last columns of the zone, and vertical and horizontal thresholds.

```
1. Project the image horizontally from top to bottom on
   the y-coordinator, from left to right on the x-
   coordinator and store it into Hor_Proj array
2. Y_top,Y_bottom = XYcut_Secondary_algorithm with the
   parameters Hor_Proj, and Vertical_Threshold
3. If size of Y_top >1
      a. For k=each row on Y_top
         i.   Project the image vertically from Y_top(k) to
              Y_bottom(k) to y-coordinator axis, from left
              to right for x-coordinator and store it into
              Ver_Proj array
         ii.  X_left,X_right = XYcut_Secondary_algorithm
              with the parameters Ver_Proj, and
              Horizontal_Threshold
         iii. For j= each row on X_left
              1) Repeat the algorithm recursively with
                 input parameters: the image, Y_top(k),
                 Y_bottom(k), X_left(j), X_right(j)
   Else
      a. Return top, left, bottom, right
```

(a)

The XY cut Secondary algorithm:

The Input Parameters: one dimensional matrix (called Project), and the Threshold.

```
1. First(1)=1
2. ind=1
3. For gap = all consecutive zero-values on Project
      a. If the size of current gap ≥ Threshold
          i.   Second(ind)= the index of the first item of
               the current gap -1
         ii.   ind=ind+1
        iii.   First (ind)= the index of the last item of
               the current gap -1
4. Return First, Second
```

(b)

Figure 15: XY cut Algorithm; (a) The XY cut main algorithm, (b) The XY cut Secondary algorithm

...اري في هرات ● مسلحون أفغان احتجزوا صحافية كندية في كهف قبل الإفراج عنها

...ي يعترف بقتله 37 مدنيا في غارة بأفغانستان

مناطق مأهولة لكي يدفعوا القوات لرد يؤدي الى مقتل مدنيين. وقال الكولونيل جريج جونيان المتحدث باسم الجيش الاميركي في البيان ناسف لهذا الفقد المأساوي لارواح ابرياء ونعبر عن تعازينا لاسر الضحايا وشعب افغانستان. وقتل نحو اربعة الاف شخص ثلثهم تقريبا من المدنيين في قتال هذا العام فيما سعد مقاتلو طالبان حملتهم للاطاحة بحكومة كـرزاي وطرد القوات الاجنبية من البلاد. وفي مدريد قال متحدث باسم الجيش الاسباني ان جنديين اسبانيين قتلا واصيب ثالث بجراح خطيرة في هجوم انتحاري امس بجنوب اقليم هرات غرب افغانستان.

وقال متحدث باسم الجيش الايطالي الذي يتولي حاليا قيادة قوات حلف شمال الأطلسي بغرب افغانستان ان الوحدة الاسبانية تقدم المساعدة لقوات الامن الافغانية في منطقة شيندند جنوب هرات. وبهذا يصل عدد الجنود الاسبان الذين قتلوا في معارك مع مقاتلي طالبان الى 25 بينما يرتفع اجمالي عدد القتلى من القوات الاجنبية الى اكثر من الف منذ الاطاحة بحكومة طالبان في اواخر عام 2001. ويوجد نحو 700 جندي اسباني في افغانستان معظمهم في اقليم هرات.

من جهة اخرى قالت صحافية كندية في تصريحات بثت امس ان افغانا خطفوها واحتجزوها في كهف لمدة اربعة اسابيع قبل الافراج عنها. وافرج عن الصحافية مبليسا فانج التي تعمل لدى هيئة الاذاعة الكندية في افغانستان اول من امس بعد خطفها قبل شهر قرب العاصمة كابل. وفي شريط فيديو بثه جهاز المخابرات الافغاني الذي تمكن من الافراج عنها كانت فانج ترتدي سترة عسكرية مموهة وقالت ايضا ان يديها وقدميها كانت مقيدة خلال اخر اسبوع من احتجازها في اقليم ميدان وردك الى الجنوب لغربي من كابل. وقالت هيئة الاذاعة الكندية في بيان ان مسلحين خطفوا فانج عند مخيم للاجئين على مشارف كابل يوم 12 اكتوبر (تشرين الاول) ونقلت الى منطقة جبلية الى الغرب من المدينة. وشهدت أفغانستان وهي واحدة من أخطر بلاد العالم بالنسبة للصحافيين تصاعدا في جرائم اغتيال وخطف الأجانب في الشهور الأخيرة. وقالت ادارة المخابرات ان فانج افرج عنها بدون اي فدية ولكنها لم تحدد ما اذا كان الخاطفون من المجرمين أم من مقاتلي طالبان الذين كانوا ايضا وراء العديد من عمليات الخطف في أفغانستان في السنوات الاخيرة.

وقال مسؤول في المخابرات ان ثلاثة رجال اعتقلوا خلال العملة ولكن أحد زعماء المجموعة فر الى الخارج.

تقرير: عدة آلاف من الإرهابيين ناشطون في بريطانيا

لندن، «الشرق الأوسط»

كشف تقرير استخباراتي حديث، عن وجود عدة الاف من العناصر الاصولية المتشددة النشطة في بريطانيا، التي تعد من الأهداف البارزة للإرهاب الدولي، كما جاء في تقرير نشر أمس.

وشارك في إعداد التقرير السري، تحت عنوان الإرهاب الدولي، الاستخبارات العسكرية التابعة لوزارة الدفاع، (الاستخبارات الداخلية) و«القسم الخاص»، وفق صحيفة «صنداي تلغراف» البريطانية، في عددها الصادر امس. وجاء في التقرير أن بريطانيا: «ستظل، وعلى المدى المنظور، في سلم أولويات أهداف الإرهابيين الدوليين المنحازين لتنظيم القاعدة.. وتواجه تهديدا من مواطنيها البريطانيين، منهم معتنقو الإسلام، وإرهابيون اجانب مقيمون بالبلاد، إلى جانب إرهابيين يخططون لضربات من خارج البلاد». ويورد التقرير أن تهديدات «المجتمع المتشدد» في المملكة المتحدة «متنوعة وواسعة الانتشار» إلا أنه يشير في ذات الوقت إلى «صعوبة تقدير» أعداد الإرهابيين في بريطانيا، وفق الصحيفة. ويقدر «مركز تحليل الإرهاب المشترك» التابع للاستخبارات البريطانية وجود «عدة الاف من المتشددين» داخل المملكة المتحدة، من الملتزمين بدعم الأنشطة الجهادية داخل أو خارج البلاد. وكان رئيس جهاز الاستخبارات القومية، جونثان إيفانز، قد أشار العام الماضي، إلى تحديد انوكالة لقرابة ألفي فرد ممن يمثلون تهديداً على الأمن القومي والسلامة العامة. وتشير إحصائية وزارة الداخلية البريطانية إلى اعتقال 1200 مشتبه بالإرهاب منذ عام 2001، تم توجيه الاتهام إلى 140 منهم، وإدانة اكثر من 45 بتهم تتصل بالإرهاب. وقدرت الوزارة وجود نحو 200 شبكة إرهابية عاملة ببريطانيا في يومنا هذا، متورطة في 30 مخططاً إرهابياً على الأقل، بحسب التقرير. وحدد التقرير الاستخباراتي تلك الفئات الارهابية بأن معظمهم من المواطنين البريطانيين من أصول آسيوية، تحديداً من باكستان، بالإضافة إلى آخرين من شمال وشرق افريقيا، والعراق والشرق الأوسط، بجانب فئة من معتنقي الإسلام حديثا.

ويذكر التقرير أن غالبية تلك الفئة المتشددة من الرجال، تتراوح أعمارهم بين سن 18 إلى 30 عاماً. وتضم تلك الفئات بعض العناصر التي تلقت تدريبات في معسكرات للإرهاب خارج بريطانيا. وتحدد أجهزة الاستخبارات البريطانية العاصمة لندن ومدينة برمنغهام وجنوب شرقي إنجلترا كمناطق نشاط لعناصر المتشددة. وتقدر أعداد المسلمين في بريطانيا بقرابة 1,5 مليون مسلم، مليون منهم يقيمون في لندن.

المذكرة النهائية في قضية القطري المري ستصدر اليوم

احتمال نظر المحكمة العليا الأميركية في شرعية اعتقال «المقاتلين الأعداء»

واشنطن، «الشرق الأوسط»

كان القطري علي صالح المري على وشك المثول أمام المحكمة بتهمة الاحتيال، عندما سعى ممثلو الادعاء إلى إحدى محاكم ولاية إلينوي بطلب إسقاط التهم لأن قرار الرئيس الأميركي جورج بوش، حسب قولهم، صنفه بأنه «مقاتل عدو». وقد اعترض محامو المري، لكن جان بول ميلر النائب العام الأميركي أعلن أن المري، الذي ينظر إليه على أنه أحد أفراد الخلايا النائمة، يخضع حاليا لسيطرة الجيش وقال «لم يعد هناك أي دعوى قضائية أمام المحكمة الآن».

بناء على ذلك نقل المري إلى سجن للبحرية في تشارلستون سي إس حيث قضى أكثر من خمس سنوات. لكن القضية تثير تساؤلا ينطوي على الكثير من الدلالات بالنسبة لسلطات الرئيس والحريات المدنية وحول سلطة الجيش في القيام بعملية احتجاز دون توجيه تهم، لمواطن أميركي أو مقيم شرعي اعتقل على الأراضي الأميركية.

وقالت صحيفة واشنطن بوست في عددها الصادر أمس: «قدمت بعض الطلبات إلى المحكمة العليا للنظر في قانونية اعتقال المري، حيث تعد قضيته واحدة من أكثر القضايا إثارة للجدل والمتعلقة بالسلطة التنفيذية منذ هجمات سبتمبر (أيلول) 2001. وبود محامو المري من المحكمة إسقاط حكم الاستئناف الذي يدعم الإدارة».

ومن المتوقع أن تصدر المذكرة النهائية اليوم، ليقرر القضاة ما إذا كان سيتم قبول القضية من عدمه. ويرد مسؤولو الإدارة الأميركية على ذلك بأن اعتقال المري ـ الذي يعتقد أنه كان يخطط لشن موجة من الهجمات في أعقاب احداث 9/11 ـ مهم لحماية الأمن القومي خلال فترة الحرب. وتقول المذكرة التي أصدروها «إن المري مثله مثل أفراد القاعدة الذين هاجموا الولايات المتحدة في 11 سبتمبر (أيلول) قدم إلى هنا لتخطيط وتنفيذ أعمال عدائية شبه حربية. ويقول محامو المري إن سلطة الاعتقال تلك غير دستورية وتثير المخاوف من إمكانية إلقاء الحكومة القبض على أحد الأفراد في الشارع، او على معارض سياسي وحبسه دون محاكمة.

يذكر أن قضية المري تحظى بمتابعة من منظمات حقوق الإنسان العالمية كمنظمتي مراقبة حقوق الإنسان لأميركية «هيومن رايتس ووتش» ومنظمة العفو الدولية «امنستي انترناشونال». وتفيد منظمة «هيومن رايتس ووتش» في أحد تقاريرها أن علي صالح المري هو مواطن قطري دخل الولايات المتحدة بتأشيرة دراسية، وألقي القبض عليه ووجهت هيئة محلفين فيدرالية الاتهام إليه بسبب ما زعم عن كذبه على المحققين، والاحتيال في استخدام البطاقات الائتمانية».

ولكن بعد صدور قرار الاتهام قررت السلطة التنفيذية أن تعيد تصنيفه من جديد على أنه من مقاتلي الأعداء، ونقلته إلى منشأة تابعة للقوات البحرية في ولاية ساوث كارولينا في 23 يونيو (حزيران) 2003. وأوضحت الحكومة أنها قررت أن المري من مقاتلي الأعداء بناء على معلومات استقتها من استجواب شخص متهم بأنه من مسؤولي تنظيم القاعدة.

(a)

ناري في هرات ● مسلحون أفغان احتجزوا صحافية كندية في كهف قبل الإفراج عنها

ي يعترف بقتله 37 مدنيا في غارة بأفغانستان

مناطق ماهولة لكي يدفعوا القوات لرد. يؤدي الى مقتل مدنيين. وقال الكولونيل جريج جوليان المتحدث باسم الجيش الاميركي في البيان ناسف لهذا الفقد المأساوي لارواح ابرياء ونعبر عن تعازينا لاسر الضحايا وشعب افغانستان. وقتل نحو اربعة الاف شخص ثلثهم تقريبا من المدنيين في قتال هذا العام فيما صعد مقاتلو طالبان حملتهم للاطاحة بحكومة كرزاي وطرد القوات الاجنبية من البلاد. وفي مدريد قال متحدث باسم الجيش

الاسباني ان جنديين اسبانيين قتلا واصيب ثالث بجراح خطيرة في هجوم انتحاري امس بجنوب اقليم هرات غرب افغانستان.

وقال متحدث باسم الجيش الايطالي الذي يتولى حاليا قيادة قوات حلف شمال الاطلسي بغرب افغانستان ان الوحدة الاسبانية تقدم المساعدة لقوات الامن الافغانية في منطقة شيندلد جنوب هرات وبهذا يصل عدد الجنود الاسبان الذين قتلوا في معارك مع مقاتلي طالبان الى 25 بينما يرتفع اجمالي عدد

القتلى من القوات الاجنبية الى اكثر من الف منذ الاطاحة بحكومة طالبان في اواخر عام 2001. ويوجد نحو 700 جندي اسباني في افغانستان معظمهم في اقليم هرات.

من جهة اخرى قالت صحافية كندية في تصريحات بثت امس ان افغانا خطفوها واحتجزوها في كهف لمدة اربعة اسابيع قبل الافراج عنها. وافرج عن الصحافية ميليسا فانج التي تعمل لدى هيئة الاذاعة الكندية في افغانستان اول من امس بعد خطفها قبل شهر قرب العاصمة

كابل. وفي شريط فيديو بثه جهاز المخابرات الافغاني الذي تمكن من الافراج عنها كانت فانج ترتدي سترة عسكرية مموهة وقالت ايضا ان يديها وقدميها كانت مقيدة خلال اخر اسبوع من احتجازها في اقليم ميدان وردك الى الجنوب الغربي من كابل. وقالت هيئة الاذاعة الكندية في بيان ان مسلحين خطفوا فانج عند مخيم للاجئين على مشارف كابل يوم 12 أكتوبر (تشرين الاول) ونقلت الى منطقة جبلية الى الغرب من المدينة. وشهدت افغانستان وهي

واحدة من أخطر بلاد العالم بالنسبة للصحافيين تصاعدا في جرائم اغتيال وخطف الاجانب في الشهور الاخيرة. وقالت ادارة المخابرات ان فانج أفرج عنها بدون أي فدية ولكنها لم تحدد ما اذا كان الخاطفون من المجرمين أم من مقاتلي طالبان الذين كانوا أيضا وراء العديد من عمليات الخطف في أفغانستان في السنوات الاخيرة.

وقال مسؤول في المخابرات ان ثلاثة رجال اعتقلوا خلال الحملة ولكن أحد زعماء المجموعة فر الى الخارج.

تقرير: عدة آلاف من الإرهابيين ناشطون في بريطانيا

لندن، «الشرق الأوسط»

كشف تقرير استخباراتي حديث، عن وجود عدة الاف من العناصر الأصولية المتشددة النشطة في بريطانيا، التي تعد من الأهداف البارزة للإرهاب الدولي، كما جاء في تقرير نشر أمس.

وشارك في إعداد التقرير السري، تحت عنوان الإرهاب الدولي، الاستخبارات العسكرية التابعة لوزارة الدفاع، (الإستخبارات الداخلية) و«القسم الخاص»، وفق صحيفه «صنداي تلغراف» البريطانية، في عددها الصادر امس. وجاء في التقرير أن بريطانيا: «سنظل، وعلى المدى المنظور، في سلم أولويات أهداف الإرهابيين الدوليين المنحازين لتنظيم القاعدة.. وتواجه تهديدا من مواطنيها البريطانيين، منهم معتنقو الإسلام، وإرهابيون أجانب مقيمون بالبلاد، إلى جانب إرهابيين يخططون لضربات من خارج البلاد».ويورد التقرير أن تهديدات «المجتمع المتشدد» في المملكة المتحدة «متنوعة وواسعة الانتشار» إلا أنه يشير في ذات الوقت إلى «صعوبة تقدير» أعداد الإرهابيين في بريطانيا، وفق الصحيفة. ويقدر «مركز تحليل الإرهاب المشترك» التابع للاستخبارات البريطانية وجود «عدة الاف من المتشددين» داخل المملكة المتحدة، من الملتزمين بدعم الأنشطة الجهادية داخل أو خارج البلاد. وكان رئيس جهاز الاستخبارات القومية، جونثان إيفانز، قد أشار العام الماضي، إلى تحديد الوكالة لقرابة ألفي فرد ممن يمثلون تهديداً على الأمن القومي والسلامة العامة. وتشير إحصائية وزارة الداخلية البريطانية إلى اعتقال 1200 مشتبه بالإرهاب منذ عام 2001، تم توجيه الاتهام إلى 140 منهم، وإدانة اكثر من 45 بتهم تتصل بالإرهاب. وقدرت الوزارة وجود نحو 200 شبكة إرهابية عاملة ببريطانيا في يومنا هذا، متورطة في 30 مخططاً إرهابياً على الأقل، بحسب التقرير. وحدد التقرير الاستخباراتي تلك الفئات الإرهابية بأن معظمهم من المواطنين البريطانيين من أصول آسيوية، تحديداً من باكستان، بالإضافة إلى آخرين من شمال وشرق افريقيا، والعراق والشرق الأوسط، بجانب فئة من معتنقي الإسلام حديثاً.

ويذكر التقرير أن غالبية تلك الفئة المتشددة من الرجال، تتراوح أعمارهم بين سن 18 إلى 30 عاماً. وتضم تلك الفئات بعض العناصر التي تلقت تدريبات في معسكرات للإرهاب خارج بريطانيا. وتحدد اجهزة الاستخبارات البريطانية العاصمة لندن ومدينة برمنغهام وجنوب شرقي إنجلترا كمناطق نشاط العناصر المتشددة. وتقدر أعداد المسلمين في بريطانيا بقرابة 1,7 مليون مسلم، مليون منهم يقيمون في لندن.

المذكرة النهائية في قضية القطري المري ستصدر اليوم

احتمال نظر المحكمة العليا الأميركية في شرعية اعتقال «المقاتلين الأعداء»

واشنطن، «الشرق الأوسط»

كان القطري علي صالح المري على وشك المثول أمام المحكمة بتهمة الاحتيال، عندما سعى ممثلو الادعاء إلى إحدى محاكم ولاية إلينوي بطلب إسقاط التهم لان قرار الرئيس الأميركي جورج بوش، حسب قولهم، صنفه بانه «مقاتل عدو». وقد اعترض محامو المري، لكن جان بول ميلر النائب العام الأميركي أعلن أن المري، الذي ينظر إليه على أنه أحد أفراد الخلايا النائمة، يخضع حاليا لسيطرة الجيش وقال «لم يعد هناك أي دعوى قضائية أمام المحكمة الآن».

بناء على ذلك نقل المري إلى سجن للبحرية في تشارلستون سي إس حيث قضى أكثر من خمس سنوات. لكن القضية تثير تساؤلا ينطوي على الكثير من الدلالات بالنسبة لسلطات الرئيس والحريات المدنية وحول سلطة الجيش في القيام بعملية احتجاز دون توجيه تهم، لمواطن أميركي أو مقيم شرعي اعتقل على الأراضي الأميركية.

وقالت صحيفة واشنطن بوست في عددها الصادر أمس: «قدمت بعض الطلبات إلى المحكمة العليا للنظر في قانونية اعتقال المري، حيث تعد قضيته واحدة من أكثر القضايا إثارة للجدل والمتعلقة بالسلطة التنفيذية منذ هجمات سبتمبر (أيلول) 2001. وبود محامو المري من المحكمة إسقاط حكم الاستئناف الذي يدعم الإدارة».

ومن المتوقع أن تصدر المذكرة النهائية اليوم، ليقرر القضاة ما إذا كان سيتم قبول القضية من عدمه ويرد مسؤولو الإدارة الأميركية على ذلك بان اعتقال المري - الذي يعتقد أنه كان يخطط لشن موجة من الهجمات في أعقاب أحداث 9/11 - مهم لحماية الأمن القومي خلال فترة الحرب. وتقول المذكرة التي أصدروها «إن المري مثله مثل أفراد القاعدة الذين هاجموا الولايات المتحدة في 11 سبتمبر (أيلول) قدم إلى هنا لتخطيط وتنفيذ أعمال عدائية شبه حربية. ويقول محامو المري إن سلطة الاعتقال تلك غير دستورية وتثير المخاوف من إمكانية إلقاء الحكومة القبض على أحد الأفراد في الشارع، أو على معارض سياسي وحبسه دون محاكمة.

يذكر أن قضية المري تحظى بمتابعة من منظمات حقوق الإنسان العالمية كمنظمتي مراقبة حقوق الإنسان الأميركية «هيومن رايتس ووتش» ومنظمة العفو الدولية «امنستي انترناشونال». وتفيد منظمة «هيومن رايتس ووتش» في أحد تقاريرها أن علي صالح المري هو مواطن قطري دخل الولايات المتحدة بتأشيرة دراسية. وألقي القبض عليه ووجهت هيئة محلفين فيدرالية الاتهام إليه بسبب ما زعم عن كذبه على المحققين، والاحتيال في استخدام البطاقات الائتمانية».

ولكن بعد صدور قرار الاتهام قررت السلطة التنفيذية أن تعيد تصنيفه من جديد على أنه من مقاتلي الأعداء، ونقله إلى منشأة تابعة للقوات البحرية في ولاية ساوث كارولينا في 23 يونيو (حزيران) 2003. وأوضحت الحكومة أنها قررت أن المري من مقاتلي الأعداء بناء على معلومات استقتها من استجواب شخص متهم بأنه من مسؤولي تنظيم القاعدة.

(b)

Figure 16: samples of applying XY cut algorithm;(a) the original image; (b) the result of applying XY cut algorithm

CHAPTER 4

SCRIPT IDENTIFICATION

4.1. Introduction

Existing script identification approaches can be global or local. In local approaches, features are extracted from a document image at the line, word or character levels. In the case of the global approaches, the features are extracted from the document at the page or text block levels. This research extracts the features at the block and word levels. In script identification, the scripts of the text regions that are extracted by document analysis and classification part is identified as Arabic or Latin scripts.

The script identification has three phases, namely, texture patch, texture feature extraction, script classification. The texture patch can be block or word texture patch. In the block texture patch, each text region is segmented into n*n block texture patches. In the word texture patch, each word of text regions is repeated horizontally and vertically to fill the word texture patch, whose height and width are standard. In the texture features extraction, features are extracted from each textures patch using Gabor filter. In script classification, a number of classifiers are applied to classify each texture patch representing a word or block to Arabic or Latin.

4.2. Block Texture Patch

Each region is divided into n*n block texture patches. Each block texture patch has the following:

1. No spaces between words/characters.
2. No spaces between text lines.
3. A standard height of its lines.

4.3. Word Texture Patch

In this phase, word texture patches are generated in order to extract texture features. Each texture patch has a standard height and width and contains a word repeated horizontally and vertically to fill it. We used the idea of normalizing text blocks presented in [Busc05]. The algorithm of building word texture patches has the following steps:

4.3.1. Lines extraction

Horizontal projection profiles are used to extract lines of the text region. The positions of line breaks are located by detecting valleys in those profiles. Figure 17 shows a sample of Latin zone with its horizontal projection.

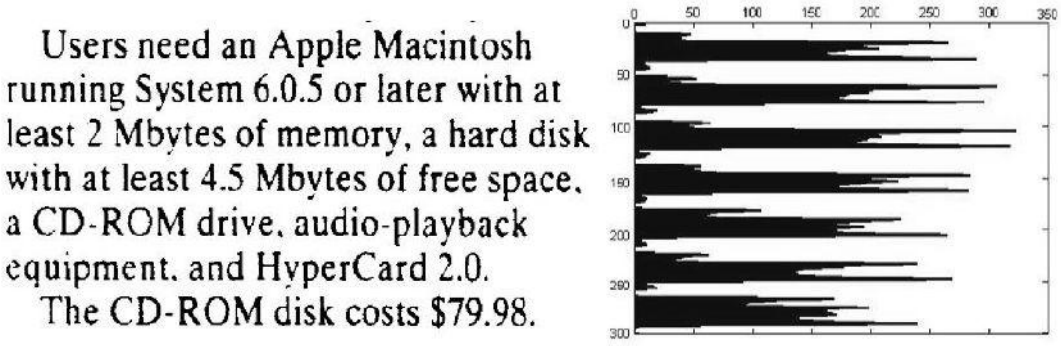


Figure 17: Examples of Latin and Arabic zones with theirs horizontal projections

4.3.2. Extracting Words

After performing the above operation, the vertical projection profiles are taken for each line. Detecting valleys in those profiles represent the spaces between words/characters. The average of the spaces between the words/characters in the line are computed. If the spaces between adjacent words/characters are smaller than the average, the adjacent words/characters are merged as one word without space. Figure 18 shows an example of extracting words of a line.

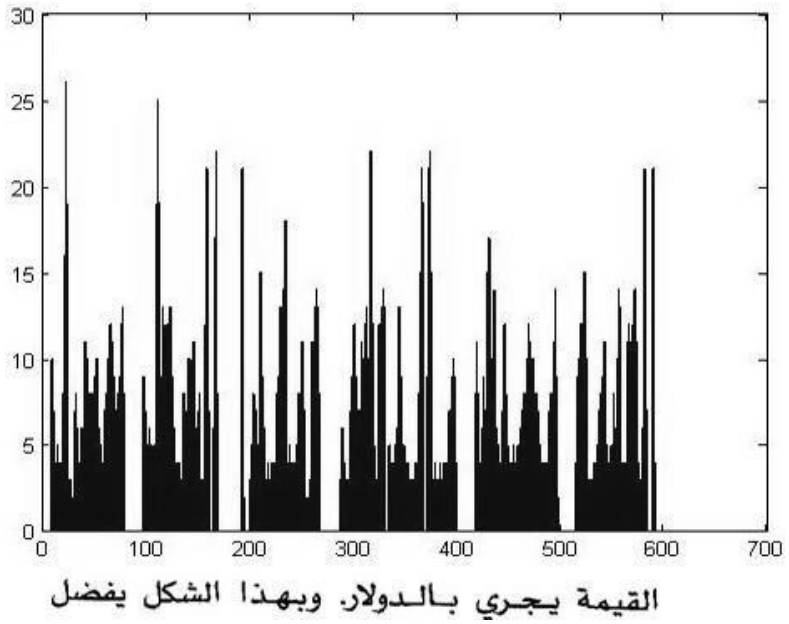


Figure 18: An Example of Extracting Words of Line

4.3.3. Normalizing Words Vertically

The height of the words is normalized to a standard height. The aspect ratio of the width to the height is kept. For example, assume word is given and its width and its height are respectively 60 and 40 pixels. The aspect ratio of the width to the height is 3/2. When the height of the word is normalized to 100 pixels, the width will be 150.

4.3.4. Generating Word Texture Patches

For each word, a word texture patch is generated with a standard height and width. Each texture patch is filled by repeating the word horizontally and vertically.

Figure 19 shows some examples of Arabic and Latin samples that are produced from applying word texture patches generation.

Figure 19 Arabic and Latin Samples of applying Text blocks Normalization

4.4. Gabor Filter

Texture features are extracted of texture patches representing words or blocks using two-dimensional Gabor filter, which is implemented in [Mahm11]. The two dimensional

Gabor filter consists of a sinusoidal plane wave modulated by a two-pass Gaussian envelope. The even and odd Gabor filters in the 2-dimensional spatial domain are formulated as:

$$g_{even}(x, y; \lambda, \theta) = e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)} \cos\left(2\pi \frac{x}{\lambda}\right) \qquad (4.1)$$

$$g_{odd}(x, y; \lambda, \theta) = e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)} \sin\left(2\pi \frac{x}{\lambda}\right) \qquad (4.2)$$

Where $\sigma_x = \lambda k_x$ and $\sigma_y = \lambda k_y$. And λ is the Gabor filter wavelength in pixels, θ is the angle of filter in degrees (zero angle gives a filter that responds to vertical features). k_x and k_y are scale factors relative to the wavelength of the filter. The bandwidth of the filter is controlled in the x-direction by k_x, whereas its orientation selectivity is controlled across the filter by k_y.

The filter orientations are calculated from:

$$\theta_k = \frac{2\pi k}{n}, k = \{0, 1, \dots, n-1\} \quad (4.3)$$

where n is the number of orientations used.

The filtering was implemented in the frequency domain in order to speed up the computation. The following process was implemented:

$$Filtered\ Image = FFT^{-1}\left[FFT\,(Image) * FFT\,(Filter)\right] \qquad (4.4)$$

Where FFT and FFT^{-1} are the fast Fourier transform and the inverse fast Fourier transform, respectively.

Six different orientations (0, 30, 60, 90, 120 and 150) and 4 scales (wavelengths of 3, 6, 12 and 24) are used.

The filtered image is segmented into n*m smaller segments (n is the number of horizontal slices and m is the number of vertical slices). The size of each segment is as follows:

$$Size\ of\ segment = \frac{Image\ Width}{n} * \frac{Image\ Height}{m}$$

The mean and variance of each segment are taken as the features of the segment. This is repeated for all filtered images at different scales, orientations, and image segments of text samples. In this research, 3*3 segments is used. This results in a feature vector of 6*4*3*3*2=432 features for 6 orientations, 4 scales, 3 horizontal segments, 3 vertical segments, and using the mean and variance.

CHAPTER 5

EXPERIMENTAL RESULTS

In this chapter, several experiments are conducted to evaluate the proposed document segmentation algorithm and compare it with existing methods.

5.1. Data and Tools

MATLAB is used to implement and test the prototype of this research. In the document analysis part, we used 398 document images selected randomly from printed Arabic text database (PATDB) which was presented in [Alha09], [Alha10a], [Alha10b]. PATDB database consists of 6954 document images selected from various printing forms (viz. advertisements, book chapters, magazines, newspapers, letters and reports). These images are stored in three different formats:

1. Black & white (binary) format with color depth of 1-bit per pixel;
2. Grayscale format with color depth of 8-bit (1-byte) per pixel (0 to 255 gray levels); and
3. Color (or RGB) format with color depth of 24-bit (3-byte) per pixel.

Each format was scanned with 200,300, and 600 dpi resolutions. Table 1 shows the number of images in the PATDB.

The database of 398 images was partitioned randomly into 120 images for testing, and 278 images for training. For validation, 55 images are selected randomly from the training set.

Table 1: THE DISTRIBUTION OF DOCUMENT IMAGES ACROSS PATDB.

Category	Color format/Resolution									Total
	Black & White			Grayscale			Color			
	200 dpi	300 dpi	600 dpi	200 dpi	300 dpi	600 dpi	200 dpi	300 dpi	600 dpi	
Advertisements	22	22	14	22	22	14	22	22	14	174
Book	111	111	111	111	111	111	0	0	0	666
Book (faxed then scanned)	111	187	111	187	111	111	0	0	0	818
Magazine	536	536	536	536	536	536	284	284	336	4120
News	35	35	35	35	35	35	0	0	0	210
Report	161	161	161	161	161	161	0	0	0	966
Letters	5	5	0	5	5	0	0	0	0	20
Total	981	1057	968	1057	981	968	306	306	350	6974

For script identification, the database is 444 pages collected from PATDB, the University of Washington English Document Image database (UW-I), and our own databases. The pages of PATDB are 202 Arabic pages (include the book chapters, magazines, newspapers, letters and reports). UW-I database [Phil93] is 190 Latin pages. Our database is 52 pages collected from book chapters. The database, which is used to identify scripts at region level, contains 7000 text samples of each script (7000 for Arabic text blocks and 7000 for Latin) and it is divided into 70% for training and 30% for testing. While the database, which is used to identify scripts at word level, contains 10000 text samples of each script for training and 5000 samples of each script for testing.

5.2. Evaluation Criteria

The evaluation criteria of page segmentation phase is different from the evaluation criteria of zone classification and script identification. For that, we have the following types of evaluation criteria:

5.2.1. Page Segmentation Evaluation Criteria

To evaluate a page segmentation algorithm, there are a number of measures that can be used [Shaf08]. In this work, two types of region-based error measures are defined.

1. Merged Zones:

   A merged zone error is a segmented zone that includes two or more zones with different ground truth values. Figure 20 shows an example of a merged zones' error. We find the overlapping areas of the merged zones (OAMZ) in our estimation. The overlapping areas of the merged zones (OAMZ) is the ratio of the overlapping areas of the merged zones (OAMZ) to the total zones.

$$\text{OAMZ} = \frac{\text{sum of the overlapping areas}}{\textit{total size of zones}} * 100$$

As shown in Figure 20.b, assume the segmented merged zone ($\boldsymbol{Z_s}$ on the Figure 20) is recognized as non text, then the equation of OAMZ is:

$$\text{OAMZ} = \frac{\boldsymbol{Z_{a2} + Z_{b1}}}{\boldsymbol{Z_a + Z_b - Z_c}} * \boldsymbol{100}$$

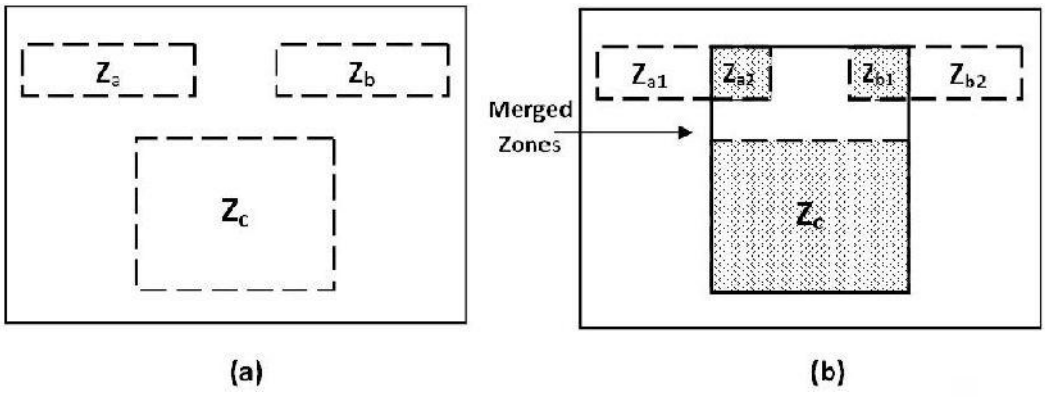


Figure 20: Merged Zones Error Measure; (a) the ground truth zones where the ground truth Z_a and Z_b are text zones and Z_c is non text. (b) a merged zone error; the shaded rectangle denotes segmented merged zone.

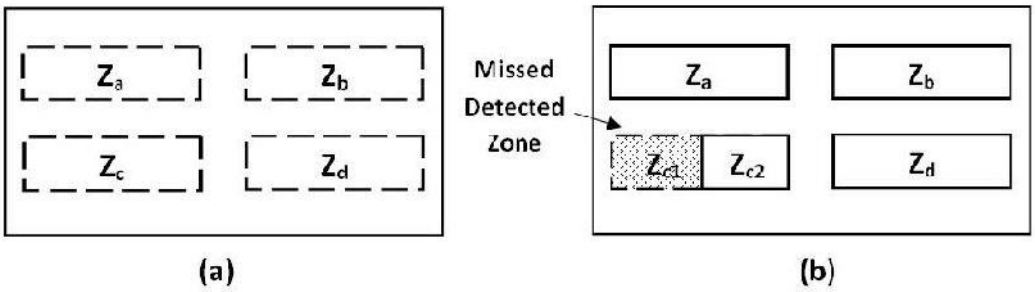


Figure 21: Missed Zones Error Measure (a) the ground truth. (b) a missed zone error; solid-line rectangles show the segmented zones while the shaded area represents the missed zone

2. Missed Zones

The missed zones are the zones that did not match any foreground zones in the hypothesized segmentation. Some zones are partially missed. So, only the missed area of those zones is counted. The following equation shows the ratio of missed areas on the missed zones to total size of zones. Figure 21 shows missed zone error.

$$Missed\ Area = \frac{missed\ areas\ on\ the\ zones}{total\ size\ of\ zones} * 100$$

The missed area error in Figure 21.b is:

$$Missed\ Area = \frac{Z_{c1}}{Z_a + Z_b + Z_d + Z_c} * 100.$$

5.2.2. Zone Classification Evaluation Criteria

Different types of metrics have been used for the performance evaluation of document analysis and classification in the classification phase. These metrics are defined below:

1. Text recognized as Text.

   The ratio of the total area of text zones in the ground truth which is recognize as text to the total area of the text zones in the ground truth.

2. Text recognized as Non Text.

   The ratio of the total area of text zones in the ground truth which is recognize as non text to the total area of the text zones in the ground truth.

3. Non Text recognized as Non Text.

   The ratio of the total area of non text zones in the ground truth which is recognize as non text to the total area of the non text zones in the ground truth.

4. Non Text recognized as Text.

   The ratio of the total area of non text zones in the ground truth which is recognize as text to the total area of the non text zones in the ground truth.

5. Percentage Accuracy

   The percentage of the Text recognized as Text, and Non Text recognized as Non Text to the total area of the zones.

$$Acc = \frac{text\ recognizd\ as\ text\ +\ non\ text\ recognized\ as\ non\ text}{Total\ area\ of\ all\ zones} * 100$$

6. Percentage of Error

   The percentage of the Text recognized as Non Text, and Non Text recognized as Text to the total area of all zones.

$$Err = \frac{text\ recognizd\ as\ non\ text\ +\ non\ text\ recognized\ as\ text}{Total\ area\ of\ all\ zones} * 100$$

5.2.3. Script Identification Evaluation Criteria

The confusion matrix, which is used to evaluate the script identification, consists of the following:

1. Arabic recognized as Arabic.

   The ratio of the number of Arabic samples in the ground truth which is recognize as Arabic to the total number of the Arabic samples in the ground truth.

2. Arabic recognized as Latin.

   The ratio of the number of Arabic samples in the ground truth which is recognize as Latin to the total number of the Arabic samples in the ground truth.

3. Latin recognized as Latin.

   The ratio of the number of Latin samples in the ground truth which is recognize as Latin to the total number of the Latin samples in the ground truth.

4. Latin recognized as Arabic.

   The ratio of the number of Latin samples in the ground truth which is recognize as Arabic to the total number of the Latin samples in the ground truth.

5. Percentage of Accuracy

   The percentage of the Arabic recognized as Arabic, and Latin recognized as Latin to the total numbers of the text samples.

$$Acc = \frac{\#\ of\ samples\ of\ Arabic\ recognized\ as\ Arabic\ +\ \#\ of\ samples\ of\ Latin\ recognized\ as\ Latin}{Total\ number\ of\ the\ text\ samples} * 100$$

6. Percentage of Error

The percentage of the Arabic recognized as Latin, and Latin recognized as Arabic to the total number of text samples.

$$Err = \frac{\#\ of\ samples\ of\ Arabic\ recognizd\ as\ Latin\ +\ \#\ of\ samples\ of\ Latin\ \ recognized\ as\ Arabic}{Total\ number\ of\ text\ blocks} * 100$$

5.3. Experimental work

In this section we present our experimentation results of the proposed document classification technique, XY cut, and RLSA methods. Each method was tested for segmentation, and zone classification.

In [Shaf08], the authors presented a performance evaluation method in order to evaluate six-page segmentation algorithms such as XY cut and RLSA. We used the same database and their method to evaluate the implemented XY cut and RLSA algorithms. Tables 2, and 3 show the results of evaluating the implemented XY cut and RLSA algorithms respectively. The error measures used are total number of correct segmentation, over segmented components, and under segmented components. The total number of correct segmentations is the total number of one-to-one matches between the ground-truth components and the segmentation components. The over segmented components are the number of ground-truth components that are divided into at least two segmented components. An under segmented component is a segmented component that includes at least two ground-truth components. The database used is University of Washington III. When comparing their implemented XY cut with our implemented XY cut, we found that the results of both were very close. The results of their implemented RLSA and our implemented were also close.

Table 2: Comparing our implemented XY Cut with Shafait et. al.'s implemented XY cut on Zone-Level Ground Truth. Where the total number of the zones are 24247.

XY Cut Algorithm	Total number of Correct Segmentations	Over segmented Components	Under segmented Components
Implemented by [Shaf08]	19.66%	11.94%	17.70%
Our im7plementation	19.2817%	12.0122%	18.3501%

Table 3: Comparing our implemented RLSA with Shafait et. al.'s implemented RLSA on Text-Line-Level Ground Truth. Where the total number of the text lines are 105443.

RLSA Algorithm	Total Correct Segmentation	Over segmented Components	Under segmented Components
Implemented by [Shaf08]	92.82%	1.64%	0.86%
Our implementation	91.735%	2.133%	1.224%

Each segmentation algorithm has its own threshold values. We experimented with different threshold values and selected those that gave the lowest error rates on the validation set . Table 4 shows the threshold values that were used for each algorithm. In the proposed segmentation algorithm, rescaling the image step divides the image into n*n window, the pixels of the window are removed if the number of foreground pixels in the window are smaller than or equal to a threshold. The lowest missed zone error is achieved when the threshold value equals 3. When the threshold is increased, the missed zone error is increased.

Table 4: Threshold Values Used for Each Algorithm in the Evaluation

Algorithm	Parameter Values
Proposed Segmentation Algorithm	`The size of the window is 5*5` `The threshold value of the number of foreground pixels in a window is 3`

RLSA	`The horizontal white run length threshold is 30.` `The Vertical white run length threshold is 28.`
XY Cut	`Horizontal and vertical thresholds are 20 and 25, respectively.`

5.3.1. Document Segmentation

Based on the performance measures defined in section 5.2.1, we evaluated the performance of the three algorithms for page segmentation.

Table 5: Comparing the Page Segmentation algorithms

Page Segmentation Algorithms	Page Segmentation Evaluation				
	Number of segmented zones	Merged Zones		Missed Detected Zones	
		Number Of Merged Zones	Ratio of the overlapping areas of the merged zones to the total size of zones	Number Of Missed Zones	the ratio of missed areas to total size of zones
Proposed	**47543**	**276**	**0.814%**	**1557**	**1.938%**
XY cut	**5846**	**279**	**1.112%**	**566**	**1.444%**
RLSA	**38744**	**424**	**1.466%**	**1220**	**1.776%**

Table 5 shows the performance of the proposed, XY cut, and RLSA algorithms. The proposed algorithm has the best performance in merged zone error measure. XY cut shows the best performance in missed zones measure. Our proposed algorithm gives the worst results in the missed zones measure. This is due to one of the steps, which is rescaling the image, leading to loss of some pixels. Rescaling the image step divides the image into n*n window, the pixels of the window are removed if the number of foreground pixels in the window are smaller than or equal to three.

5.3.2. Zone Classification

The classifiers, which were used to classify each region into text and non text, are:

1. Neural Network (Multilayer Perceptron- Back propagation)

   The Neural Network contains one hidden layer of six neurons. The transfer functions of the hidden layer and output layer are is the Log-sigmoid and Linear transfer functions respectively.

2. K- nearest Neighbor with K=1

3. Support Vector Machine (SVM)

   We used Support Vector Machine (SVM) provided by [Sher03].The SVM's kernel function is sigmoid function.

Twenty six features were extracted from each region. We evaluated each feature independently. The initial information of zone classification (viz. number of pages, number of zones, and etc.) are presented in Table 6. Tables 7,8, and 9 show the results of each feature in the proposed, XY cut, and RLSA approaches respectively using the Neural network classifier. Some features have better accuracy in some approaches. For example the "the Mean of Horizontal Projection" feature has good accuracy for XY cut and low accuracy for the proposed and RLSA approaches. The best feature in the proposed approach is 'Background Vertical Run Length Mean'. In the case of XY cut approach the best accuracy is achieved with 'Foreground Right-diagonal Run Length Variance'. While the 'Foreground Left-diagonal Run Length Mean' feature is the best for the RLSA approach. As shown in Table 7, the features, whose accuracy are zero in the proposed approach, are the circularity, the foreground horizontal run length variance, the foreground left-diagonal run length variance, and the foreground

right-diagonal run length variance. As shown in Table 8, the features, whose accuracy are zero in the XY cut approach, are the number of connected components, the background horizontal run length variance, and the aspect ratio. As shown in Table 9, the features, whose accuracy are zero in the RLSA approach, are the mean of horizontal projection, the foreground horizontal run length variance, the foreground vertical run length variance, the foreground left-diagonal run length variance, and the foreground right-diagonal run length variance.

Table 6: The initial information of Zone Classification

Attributes	Approaches			Ground Truth
	Proposed	XY Cut	RLSA	
Number of Pages	398	398	398	398
Number of Train Pages	278	278	278	278
Number of Test Pages	120	120	120	120
Total Num. of Zones	47543	5846	38744	6074
Num. of Text Zones	33889	5061	28285	5111
Num. of Non Text Zones	13378	506	10035	963
Num. of Merged Zones	276	279	424	--
Num. of Text Train Zones	22593	3543	18857	3570
Num. of Non Text Train Zones	8919	354	6690	661
Num. of Non Text Test Zones	4459	152	3345	302
Num. of Text Test Zones	11296	1518	9428	1541

Table 7: Results of each feature of the proposed approach using the NN classifier. The features are ranked (best at top)

Feature Name	Feature No	Correct Classification				Misclassification				Accuracy %	Error %
		Text / Text		Non Text / Non Text		Text / Non Text		Non Text / Text			
		Area	%	Area	%	Area	%	Area	%		
Back. Ver. RL. Mean	**Feat.15**	8139196	95.123	7182201	96.022	417274	4.877	297579	3.978	95.542	4.458
Stand. Dev. of fore.	**Feat.1**	8112140	94.807	7189524	96.119	444330	5.193	290256	3.881	95.419	4.581
Back. Left-diag. RL Variance	**Feat.20**	8326327	97.310	6971037	93.198	230143	2.690	508743	6.802	95.392	4.608
Foreground Mean	**Feat.3**	8051506	94.098	7245188	96.864	504964	5.902	234592	3.136	95.388	4.612
Back. Hor. RL Mean	**Feat.11**	8458313	98.853	6831637	91.335	98157	1.147	648143	8.665	95.346	4.654
Background Mean	**Feat.4**	8040136	93.966	7247653	96.897	516334	6.034	232127	3.103	95.333	4.667
Back. Right-diag. RL Variance	**Feat.24**	8171802	95.504	7087345	94.753	384668	4.496	392435	5.247	95.154	4.846
Back. Hor. RL Variance	**Feat.12**	8458313	98.853	6799073	90.899	98157	1.147	680707	9.101	95.143	4.857
Back. Ver. RL Variance	**Feat.16**	8445990	98.709	6710420	89.714	110480	1.291	769360	10.286	94.513	5.487
Back. Right-diag. RL Mean	**Feat.23**	7669156	89.630	7262536	97.096	887314	10.370	217244	2.904	93.112	6.888
Back. Left-diag. RL Mean	**Feat.19**	7644971	89.347	7274386	97.254	911499	10.653	205394	2.746	93.035	6.965
# of Con. Comp.	**Feat.2**	8488069	99.201	5632926	75.309	68401	0.799	1846854	24.691	88.057	11.943
Fore. Hor. RL Mean	**Feat.9**	6055136	70.767	5865490	78.418	2501334	29.233	1614290	21.582	74.335	25.665
Aspect Ratio	**Feat.5**	3874195	45.278	7246911	96.887	4682275	54.722	232869	3.113	69.350	30.650
Mean of Ver. Proj.	**Feat.8**	4602532	53.790	6417562	85.799	3953938	46.210	1062218	14.201	68.720	31.280
Fore. Right-diag. RL Mean	**Feat.21**	8100953	94.676	1977607	26.439	455517	5.324	5502173	73.561	62.849	37.151
Fore. Left-diag. RL Mean	**Feat.17**	8204950	95.892	1760999	23.543	351520	4.108	5718781	76.457	62.146	37.854
Area of Large Hor. Blank Blocks	**Feat.25**	7513995	87.817	2004251	26.796	1042475	12.183	5475529	73.204	59.355	40.645
Area of Large Ver. Blank Blocks	**Feat.26**	6332385	74.007	2801905	37.460	2224085	25.993	4677875	62.540	56.960	43.040

Mean of Hor. Proj.	**Feat.7**	8134241	95.065	303158	4.053	422229	4.935	7176622	95.947	52.615	47.385
Fore. Ver. RL Variance	**Feat.14**	236	0.003	7477624	99.971	8556234	99.997	2156	0.029	46.631	53.369
Fore. Ver. RL Mean	**Feat.13**	2260066	26.414	4946944	66.138	6296404	73.586	2532836	33.862	44.942	55.058
Circularity	**Feat.6**	0	0.000	7479780	100.000	8556470	100	0	0.000	46.643	53.357
Fore. Hor. RL Variance	**Feat.10**	0	0.000	7479780	100.000	8556470	100	0	0.000	46.643	53.357
Fore. Left-diag. RL Variance	**Feat.18**	0	0.000	7479780	100.000	8556470	100	0	0.000	46.643	53.357
Fore. Right-diag. RL Variance	**Feat.22**	0	0.000	7479780	100.000	8556470	100	0	0.000	46.643	53.357

Table 8: : Results of each feature of the XY Cut approach using the NN classifier. The features are ranked (best at top)

Feature Name	**Feature No**	**Correct Classification**				**Misclassification**				**Accuracy %**	**Error %**
		Text / Text		**Non Text / Non Text**		**Text / Non Text**		**Non Text / Text**			
		Area	**%**	**Area**	**%**	**Area**	**%**	**Area**	**%**		
Fore. Right-diag. RL Variance	**Feat.22**	168853252	99.670	49063888	90.791	559870	0.330	4976368	9.209	97.522	2.478
Fore. Ver. RL Variance	**Feat.14**	168699658	99.579	47875456	88.592	713464	0.421	6164800	11.408	96.922	3.078
Fore. Left-diag. RL Variance	**Feat.18**	168853252	99.670	47487496	87.874	559870	0.330	6552760	12.126	96.817	3.183
Fore. Right-diag. RL Mean	**Feat.21**	168600744	99.520	41629920	77.035	812378	0.480	12410336	22.965	94.083	5.917
Fore. Left-diag. RL Mean	**Feat.17**	168665508	99.559	41136612	76.122	747614	0.441	12903644	23.878	93.891	6.109
Fore. Ver. RL Mean	**Feat.13**	168412837	99.410	41103579	76.061	1000285	0.590	12936677	23.939	93.763	6.237
Fore. Hor. RL Variance	**Feat.10**	168779735	99.626	38807810	71.813	633387	0.374	15232446	28.187	92.900	7.100
Fore. Hor. RL Mean	**Feat.9**	168732669	99.598	38807810	71.813	680453	0.402	15232446	28.187	92.879	7.121
Mean of Hor. Proj.	**Feat.7**	157731746	93.105	41999873	77.720	11681376	6.895	12040383	22.280	89.384	10.616
Back. Ver. RL Variance	**Feat.16**	169397786	99.991	29109160	53.866	15336	0.009	24931096	46.134	88.836	11.164
Back. Right-diag. RL Variance	**Feat.24**	164544993	97.126	31556370	58.394	4868129	2.874	22483886	41.606	87.759	12.241

Back. Left-diag. RL Variance	**Feat.20**	164544993	97.126	26193166	48.470	4868129	2.874	27847090	51.530	85.359	14.641
Foreground Mean	**Feat.3**	169289553	99.927	18196826	33.673	123569	0.073	35843430	66.327	83.904	16.096
Background Mean	**Feat.4**	169267827	99.914	18203183	33.684	145295	0.086	35837073	66.316	83.897	16.103
Back. Ver. RL. Mean	**Feat.15**	166494096	98.277	13768278	25.478	2919026	1.723	40271978	74.522	80.671	19.329
Back. Left-diag. RL Mean	**Feat.19**	166501787	98.282	13215848	24.456	2911335	1.718	40824408	75.544	80.427	19.573
Back. Right-diag. RL Mean	**Feat.23**	166501787	98.282	13215848	24.456	2911335	1.718	40824408	75.544	80.427	19.573
Back. Hor. RL Mean	**Feat.11**	168133954	99.245	6925613	12.816	1279168	0.755	47114643	87.184	78.343	21.657
Stand. Dev. of fore.	**Feat.1**	168448211	99.430	5578974	10.324	964911	0.570	48461282	89.676	77.881	22.119
Back Pixels / Total Pixels	**Feat.25**	164139910	96.887	477920	0.884	5273212	3.113	53562336	99.116	73.670	26.330
Mean of Ver. Proj.	**Feat.8**	110528600	65.242	50713151	93.843	58884522	34.758	3327105	6.157	72.159	27.841
Area of Large Ver. Blank Blocks	**Feat.26**	128020163	75.567	25738039	47.628	41392959	24.433	28302217	52.372	68.810	31.190
Circularity	**Feat.6**	140361454	82.852	25943	0.048	29051668	17.148	54014313	99.952	62.826	37.174
# of Con. Comp.	**Feat.2**	**169413122**	**100**	**0**	**0.000**	**0**	**0.000**	**54040256**	**100**	**75.816**	**24.184**
Back. Hor. RL Variance	**Feat.12**	**169413122**	**100**	**0**	**0.000**	**0**	**0.000**	**54040256**	**100**	**75.816**	**24.184**
Aspect Ratio	**Feat.5**	**169397082**	**99.991**	**0**	**0.000**	**16040**	**0.009**	**54040256**	**100**	**75.809**	**24.191**

Table 9: Results of each feature of the RLSA approach using the NN classifier. The features are ranked (best at top)

Feature Name	Feature No	Correct Classification				Miss classification				Accuracy %	Error %
		Text / Text		Non Text / Non Text		Text / Non Text		Non Text / Text			
		Area	%	Area	%	Area	%	Area	%		
Fore. Left-diag. RL Mean	**Feat.17**	189109339	95.594	90088055	92.440	8717065	4.406	7367379	7.560	94.553	5.447
Fore. Hor. RL Mean	**Feat.9**	195116797	98.630	83902932	86.094	2709607	1.370	13552502	13.906	94.493	5.507
Fore. Right-diag. RL Mean	**Feat.21**	189570161	95.827	88365288	90.673	8256243	4.173	9090146	9.327	94.125	5.875
Fore. Ver. RL Mean	**Feat.13**	188946632	95.511	81192013	83.312	8879772	4.489	16263421	16.688	91.485	8.515
Background Mean	**Feat.4**	194277785	98.206	73199595	75.111	3548619	1.794	24255839	24.889	90.584	9.416
Back. Left-diag. RL Variance	**Feat.20**	194063069	98.098	70086887	71.917	3763335	1.902	27368547	28.083	89.457	10.543
Back. Right-diag. RL Variance	**Feat.24**	195705176	98.928	64707778	66.397	2121228	1.072	32747656	33.603	88.191	11.809
Back. Ver. RL Variance	**Feat.16**	191769736	96.938	62465621	64.097	6056668	3.062	34989813	35.903	86.099	13.901
Foreground Mean	**Feat.3**	193250491	97.687	48564337	49.832	4575913	2.313	48891097	50.168	81.893	18.107
Back. Right-diag. RL Mean	**Feat.23**	196349652	99.254	33037179	33.900	1476752	0.746	64418255	66.100	77.684	22.316
Back. Ver. RL. Mean	**Feat.15**	195976685	99.065	32260003	33.102	1849719	0.935	65195431	66.898	77.295	22.705
Back. Left-diag. RL Mean	**Feat.19**	194935560	98.539	29247235	30.011	2890844	1.461	68208199	69.989	75.922	24.078
Stand. Dev. of fore.	**Feat.1**	197162614	99.664	22751001	23.345	663790	0.336	74704433	76.655	74.476	25.524
Aspect Ratio	**Feat.5**	127115965	64.256	91063427	93.441	70710439	35.744	6392007	6.559	73.889	26.111
Area of Large Ver. Blank Blocks	**Feat.26**	163740769	82.770	48971566	50.250	34085635	17.230	48483868	49.750	72.037	27.963
Back. Hor. RL Mean	**Feat.11**	197425767	99.797	5878291	6.032	400637	0.203	91577143	93.968	68.851	31.149
Mean of Ver. Proj.	**Feat.8**	115764847	58.518	85270283	87.497	82061557	41.482	12185151	12.503	68.082	31.918
# of Con. Comp.	**Feat.2**	197653559	99.913	299671	0.307	172845	0.087	97155763	99.693	67.039	32.961

Back. Hor. RL Variance	**Feat.12**	197711889	99.942	39841	0.041	114515	0.058	97415593	99.959	66.971	33.029
Circularity	**Feat.6**	156534788	79.127	31482522	32.305	41291616	20.873	65972912	67.695	63.674	36.326
Area of Large Hor. Blank Blocks	**Feat.25**	89194900	45.087	21622157	22.187	1.09E+08	54.913	75833277	77.813	37.529	62.471
Mean of Hor. Proj.	**Feat.7**	0	0.000	97455434	100	1.98E+08	100	0	0.000	33.004	66.996
Fore. Hor. RL Variance	**Feat.10**	0	0.000	97455434	100	1.98E+08	100	0	0.000	33.004	66.996
Fore. Ver. RL Variance	**Feat.14**	0	0.000	97455434	100	1.98E+08	100	0	0.000	33.004	66.996
Fore. Left-diag. RL Variance	**Feat.18**	0	0.000	97455434	100	1.98E+08	100	0	0.000	33.004	66.996
Fore. Right-diag. RL Variance	**Feat.22**	0	0.000	97455434	100	1.98E+08	100	0	0.000	33.004	66.996

5.3.2.1. Feature Selection

We implemented 26 features. Each features is evaluated independently. We selected the features with high accuracy. Based on the results of Tables 7,8, and 9, we selected the features with high accuracy as shown on Table 10.

Table 10: The proposed, XY cut, and RLSA approaches selected features

Proposed Approach		XY Cut Approach		RLSA Approach	
Feature Name	Feature No	Feature Name	Feature No	Feature Name	Feature No
Standard deviation of foreground	Featur1	Mean of Horizontal Projection	Feature7	Mean of Foreground	Featur3
Number of Connected Components	Featur2	Foreground Horizontal Run Length Mean	Feature9	Mean of Background	Featur4
Foreground Mean	Featur3	Foreground Horizontal Run Length Variance	Feature10	Foreground Horizontal Run Length Mean	Feature9
Background Mean	Featur4	Foreground Vertical Run Length Mean	Feature13	Foreground Vertical Run Length Mean	Feature13
Background Horizontal Run Length Mean	Featur11	Foreground Vertical Run Length Variance	Feature14	Background Vertical Run Length Variance	Feature16
Background Horizontal Run Length Variance	Featur12	Background Vertical Run Length Variance	Feature16	Foreground Left-diag. Run Length Mean	Feature17
Background Vertical Run Length Mean	Featur15	Foreground Left-diag. Run Length Mean	Feature17	Background Left-diag. Run Length Variance	Feature20
Background Vertical Run Length Variance	Featur16	Foreground Left-diag. Run Length Variance	Feature18	Foreground Right-diag. Run Length Mean	Feature21
Background Left-diag. Run Length Mean	Featur19	Background Left-diag. Run Length Variance	Feature20	Background Right-diag. Run Length Variance	Featur24
Background Left-diag. Run Length Variance	Featur20	Foreground Right-diag. Run Length Mean	Feature21		
Background Right-diag. Run Length Mean	Featur23	Foreground Right-diag. Run Length Variance	Feature22		
Background Right-diag. Run Length Variance	Featur24	Background Right-diag. Run Length Variance	Featur24		

For feature selection, we used the neural network classifier. We followed the following process:

a. Sequential Forward Features Selection (SFFS).

Sequential Forward Features Selection (SFFS) starts with a candidate feature subset that is the empty set. Then it iteratively adds one feature at a time to the candidate feature subset, as long as the measure of performance is improved.

As shown in Table 12 , the selected features of the proposed approach, that achieved the highest accuracy, are:

1. The Number of Connected Components
2. The Background Mean
3. Background horizontal Run Length Mean
4. Background horizontal Run Length Variance.
5. Background Vertical Run Length Mean.
6. Background Left-diagonal Run Length Mean.
7. Background Left-diagonal Run Length Variance.
8. Background Right-diagonal Run Length Mean.

As shown in Table 13, the selected features of XY cut approach, which give the best accuracy, consists of the following 9 features:

1. Foreground horizontal Run Length Mean.
2. Foreground horizontal Run Length Variance.
3. Foreground Vertical Run Length Variance.
4. Background Vertical Run Length Variance.
5. Foreground Left-diagonal Run Length Variance.
6. Background Left-diagonal Run Length Variance.
7. Foreground Right-diagonal Run Length Mean.
8. Foreground Right-diagonal Run Length Variance.
9. Background Right-diagonal Run Length Variance.

According to Table 14, the best selected features of RLSA approach are as following:

1. The Foreground Mean
2. Foreground Horizontal Run Length Mean.
3. Foreground Vertical Run Length Mean.
4. Background Vertical Run Length Variance.
5. Foreground Left-diagonal Run Length Mean.
6. Background Left-diagonal Run Length Variance.
7. Foreground Right-diagonal Run Length Mean.
8. Background Right-diagonal Run Length Variance.

The best recognition rates for our proposed, XY cut, and RLSA algorithms are respectively 98.206%, 98.180%, and 98.158%.

b. Sequential Backward Features Selection (SBFS)

The algorithm starts with all features. Then it repeatedly removes the feature whose removal yields the maximal performance improvement.

Tables 15, 16, and 17 show the features selection of our proposed, XY Cut, and RLSA approaches respectively based on SBFS method.

Using the proposed approach, the best accuracy is achieved with the following features:

1. Standard deviation of foreground
2. Number of Connected Components
3. Background Mean
4. Background Horizontal Run Length Mean
5. Background Horizontal Run Length Variance

6. Background Vertical Run Length Mean.
7. Background Vertical Run Length Variance.
8. Background Left-diagonal Run Length Mean .
9. Background Left-diagonal Run Length Variance.
10. Background Right-diagonal Run Length Mean.
11. Background Right-diagonal Run Length Variance.

Using the XY cut approach , the best selected features are the following:

1. Foreground horizontal Run Length Mean
2. Foreground horizontal Run Length Variance .
3. Foreground Vertical Run Length Mean.
4. Background Vertical Run Length Variance .
5. Foreground Right-diagonal Run Length Variance.

The best selected feature for RLSA has the following 5 features:

1. Background Mean
2. Foreground horizontal Run Length Mean.
3. Foreground Vertical Run Length Mean
4. Background Left-diagonal Run Length Variance.
5. Background Right-diagonal Run Length Variance .

The best accuracy for our proposed algorithm, XY cut, and RLSA are respectively 98.156%, 98.759%, and 96. 946%.

We evaluated the best features, which are selected by SFFS and SBFS, by using K-Nearest Neighbor and Support Vector Machine classifiers. Table 18 shows the results of

using K-NN and SVM classifiers to evaluate the best features selected by SFFS. Table 19 shows the results of using K-NN and SVM classifiers to evaluate the best features selected by SBFS. For the proposed algorithm, the accuracies of the selected features using SFFS are 97.241% and 94.019% with K-NN and SVM respectively. The accuracies of the selected features using SBFS are 96.788% and 93.891% with K-NN and SVM respectively. For the XY cut algorithm, the accuracies of the selected features using SFFS are 91.003% and 92.859% with K-NN and SVM respectively. The accuracies of the selected features using SBFS are 87.502% and 92.844% with K-NN and SVM respectively. For the RLSA algorithm, the accuracies of the selected features using SFFS are 93.925% and 92.703% with K-NN and SVM respectively. The accuracies of the selected features using SBFS are 94.577% and 91.822% with K-NN and SVM respectively.

In general, the proposed algorithm shows the best recognition rates in the merged zones error, the case of using SFFS method with NN, K-NN, and SVM classifiers, and the case of using SBFS method with K-NN and SVM classifiers. The proposed algorithm shows better recognition rate than the RLSA algorithm with NN classifier when using SBFS method to select features. While it shows the worst performance in the missed areas error because of one of its steps leading to loss of some pixels. This step is rescaling the image. The XY cut algorithm shows the best performance in two case, namely, the missed zone error and the case of using SBFS method with NN classifier. Table 11 shows the summary of evaluating the proposed, XY cut and RLSA algorithms.

Table 11: Summary Result of evaluating the proposed algorithm compared with XY cut and RLSA

The Algorithms	Segmentation		Zone Classification					
	Overlapping areas in the merged zones	the missed areas error	Selected features rate using SFFS			Selected features rate using SBFS		
			NN	K-NN	SVM	NN	K-NN	SVM
Proposed Algorithm	**0.814%**	1.938%	**98.206%**	**97.241%**	**94.019%**	98.156%	**96.788%**	**93.891%**
XY Cut	1.112%	**1.444%**	98.180%	91.003%	92.859%	**98.759%**	87.502%	92.844%
RLSA	1.466%	1.776%	98.158%	93.925%	92.703%	96.946%	94.577%	91.822%

Table 12: The results of the features selection on the proposed approach based on SFFS method using the NN classifier

Features Name	Correct Classification				Miss classification				Accuracy %	Error %
	Text / Text		Non Text / Non Text		Text / Non Text		Non Text / Text			
	Area	%	Area	%	Area	%	Area	%		
Back. Ver. RL. Mean+ # of Con. Comp	8484391	99.158	7173625	95.907	72079	0.842	306155	4.093	97.641	2.359
Back. Ver. RL. Mean+ Stand. Dev. of fore.	8353278	97.625	7187048	96.086	203192	2.375	292732	3.914	96.907	3.093
Back. Ver. RL. Mean + # of Con. Comp+ Back. Left-diag. RL Mean	8485404	99.169	7173211	95.901	71066	0.831	306569	4.099	97.645	2.355
Back. Ver. RL. Mean+ # of Con. Comp + Back. Right-diag. RL Variance	8473205	99.027	7184031	96.046	83265	0.973	295749	3.954	97.637	2.363
Back. Ver. RL. Mean + # of Con. Comp+ Back. Left-diag. RL Mean + Back. Hor. RL Variance	8486282	99.180	7235189	96.730	70188	0.820	244591	3.270	98.037	1.963
Back. Ver. RL. Mean + # of Con. Comp+ Back. Left-diag. RL Mean + Back. Left-diag. RL Variance	8471578	99.008	7188747	96.109	84892	0.992	291033	3.891	97.656	2.344
Back. Ver. RL. Mean + # of Con. Comp+ Back. Left-diag. RL Mean + Back. Hor. RL Variance + Background Mean	8471913	99.012	7189626	96.121	84557	0.988	290154	3.879	97.663	2.337
Back. Ver. RL. Mean + # of Con. Comp+ Back. Left-diag. RL Mean + Back. Hor. RL Variance + Back. Hor. RL Mean	8474509	99.042	7165920	95.804	81961	0.958	313860	4.196	97.532	2.468
Back. Ver. RL. Mean + # of Con. Comp+ Back. Left-diag. RL Mean + Back. Hor. RL Variance + Background Mean+ Back. Left-diag. RL Variance	8472622	99.020	7248407	96.907	83848	0.980	231373	3.093	98.034	1.966
Back. Ver. RL. Mean + # of Con. Comp+ Back. Left-diag. RL Mean + Back. Hor. RL Variance + Background Mean+ Back. Right-diag. RL Mean	8480833	99.116	7234572	96.722	75637	0.884	245208	3.278	97.999	2.001
Back. Ver. RL. Mean + # of Con. Comp+ Back. Left-diag. RL Mean + Back. Hor. RL Variance + Background Mean+ Back. Left-diag. RL Variance+ Back. Hor. RL Mean	8463621	98.915	7236815	96.752	92849	1.085	242965	3.248	97.906	2.094
Back. Ver. RL. Mean + # of Con. Comp+ Back. Left-diag. RL Mean + Back. Hor. RL Variance + Background Mean+ Back. Left-diag. RL Variance+ Stand. Dev. of fore.	8479557	99.101	7216208	96.476	76913	0.899	263572	3.524	97.877	2.123
Back. Ver. RL. Mean + # of Con. Comp+ Back. Left-diag. RL Mean + Back. Hor. RL Variance + Background Mean+ Back. Left-diag. RL Variance+ Back. Hor. RL Mean+ Back. Right-diag. RL Mean	**8483884**	**99.152**	**7264696**	**97.124**	**72586**	**0.848**	**215084**	**2.876**	**98.206**	**1.794**

Back. Ver. RL. Mean + # of Con. Comp+ Back. Left-diag. RL Mean + Back. Hor. RL Variance + Background Mean+ Back. Left-diag. RL Variance+ Back. Hor. RL Mean+ Stand. Dev. of fore.	8454648	98.810	7253065	96.969	101822	1.190	226715	3.031	97.951	2.049
Back. Ver. RL. Mean + # of Con. Comp+ Back. Left-diag. RL Mean + Back. Hor. RL Variance + Background Mean+ Back. Left-diag. RL Variance+ Back. Hor. RL Mean+ Back. Right-diag. RL Mean+ Back. Ver. RL Variance	8455542	98.820	7253964	96.981	100928	1.180	225816	3.019	97.962	2.038
Back. Ver. RL. Mean + # of Con. Comp+ Back. Left-diag. RL Mean + Back. Hor. RL Variance + Background Mean+ Back. Left-diag. RL Variance+ Back. Hor. RL Mean+ Back. Right-diag. RL Mean+ Back. Ver. RL Variance+ Back. Right-diag. RL Variance	8461197	98.887	7258310	97.039	95273	1.113	221470	2.961	98.025	1.975
Back. Ver. RL. Mean + # of Con. Comp+ Back. Left-diag. RL Mean + Back. Hor. RL Variance + Background Mean+ Back. Left-diag. RL Variance+ Back. Hor. RL Mean+ Back. Right-diag. RL Mean+ Back. Ver. RL Variance+ Back. Right-diag. RL Variance+ Stand. Dev. of fore.+ Foreground Mean	8469384	98.982	7243172	96.837	87086	1.018	236608	3.163	97.981	2.019

Table 13: The results of the features selection on the XY approach based on SFFS method using the NN classifier

Features Name	**Correct Classification**				**Miss classification**				**Accuracy %**	**Error %**
	Text / Text		**Non Text / Non Text**		**Text / Non Text**		**Non Text / Text**			
	Area	**%**	**Area**	**%**	**Area**	**%**	**Area**	**%**		
Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean	168755024	99.61	49343460	91.309	658098	0.388	4696796	8.691	97.604	2.396
Fore. Right-diag. RL Variance+ Fore. Ver. RL Variance	168848438	99.67	49116784	90.889	564684	0.333	4923472	9.111	97.544	2.456
Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Hor. RL Mean	168842374	99.66	49557750	91.705	570748	0.337	4482506	8.295	97.739	2.261
Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Back. Ver. RL Variance	168665508	99.56	48970130	90.618	747614	0.441	5070126	9.382	97.396	2.604
Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Variance	168837560	99.66	49557750	91.705	575562	0.340	4482506	8.295	97.736	2.264
Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Hor. RL Mean+ Fore. Left-diag. RL Mean	168914878	99.71	47981358	88.788	498244	0.294	6058898	11.212	97.066	2.934
Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Variance + Back. Right-diag. RL Variance	168909738	99.70	47908362	88.653	503384	0.297	6131894	11.347	97.031	2.969

Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Variance + Fore. Hor. RL Variance	168909986	99.70	47753093	88.366	503136	0.297	6287163	11.634	96.961	3.039
Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Variance + Back. Right-diag. RL Variance+ Fore. Left-diag. RL Variance	168836940	99.66	49616872	91.815	576182	0.340	4423384	8.185	97.763	2.237
Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Variance + Back. Right-diag. RL Variance+ Back. Left-diag. RL Variance	168910064	99.70	49052161	90.770	503058	0.297	4988095	9.230	97.543	2.457
Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Variance + Back. Right-diag. RL Variance+ Fore. Left-diag. RL Variance+ Back. Left-diag. RL Variance	168918569	99.71	50027823	92.575	494553	0.292	4012433	7.425	97.983	2.017
Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Variance + Back. Right-diag. RL Variance+ Fore. Left-diag. RL Variance+ Fore. Hor. RL Variance	168909986	99.70	47906089	88.649	503136	0.297	6134167	11.351	97.030	2.970
Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Variance + Back. Right-diag. RL Variance+ Fore. Left-diag. RL Variance+ Back. Left-diag. RL Variance+ Fore. Hor. RL Variance	167753577	99.02	50282965	93.047	1659545	0.980	3757291	6.953	97.576	2.424
Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Variance + Back. Right-diag. RL Variance+ Fore. Left-diag. RL Variance+ Back. Left-diag. RL Variance+ Fore. Left-diag. RL Mean	168918569	99.71	49113273	90.883	494553	0.292	4926983	9.117	97.574	2.426
Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Variance + Back. Right-diag. RL Variance+ Fore. Left-diag. RL Variance+ Back. Left-diag. RL Variance+ Fore. Hor. RL Variance+ Back. Ver. RL Variance	**168678498**	**99.57**	**50707516**	**93.833**	**734624**	**0.434**	**3332740**	**6.167**	**98.180**	**1.820**
Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Variance + Back. Right-diag. RL Variance+ Fore. Left-diag. RL Variance+ Back. Left-diag. RL Variance+ Fore. Hor. RL Variance+ Back. Ver. RL Variance+ Fore. Left-diag. RL Mean	168837560	99.66	49557750	91.705	575562	0.340	4482506	8.295	97.736	2.264
Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Variance + Back. Right-diag. RL Variance+ Fore. Left-diag. RL Variance+ Back. Left-diag. RL Variance+ Fore. Hor. RL Variance+ Back. Ver. RL Variance+ Fore. Left-diag. RL Mean+ Fore. Ver. RL Mean	168826334	99.65	49223111	91.086	586788	0.346	4817145	8.914	97.582	2.418

Table 14: The results of the features selection on the RLSA approach based on SFFS method using the NN classifier

Features Name	Correct Classification				Miss classification				Accuracy %	Error %
	Text / Text		Non Text / Non Text		Text / Non Text		Non Text / Text			
	Area	%	Area	%	Area	%	Area	%		
Fore. Left-diag. RL Mean+ Fore. Hor. RL Mean	191191861	96.646	89291053	91.622	6634543	3.354	8164381	8.378	94.988	5.012
Fore. Left-diag. RL Mean+ Back. Ver. RL Variance	185832866	93.937	93212084	95.646	11993538	6.063	4243350	4.354	94.501	5.499
Fore. Left-diag. RL Mean+ Back. Right-diag. RL Variance	191758674	96.933	84870817	87.087	6067730	3.067	12584617	12.913	93.683	6.317
Fore. Left-diag. RL Mean+ Fore. Hor. RL Mean+ Back. Left-diag. RL Variance	195481591	98.815	92969192	95.397	2344813	1.185	4486242	4.603	97.687	2.313
Fore. Left-diag. RL Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Mean	187073062	94.564	89140021	91.467	10753342	5.436	8315413	8.533	93.542	6.458
Fore. Left-diag. RL Mean+ Fore. Hor. RL Mean+ Back. Left-diag. RL Variance+ Back. Ver. RL Variance	193609801	97.869	93558022	96.001	4216603	2.131	3897412	3.999	97.252	2.748
Fore. Left-diag. RL Mean+ Fore. Hor. RL Mean+ Back. Left-diag. RL Variance+ Back. Right-diag. RL Variance	191901607	97.005	93550699	95.993	5924797	2.995	3904735	4.007	96.671	3.329
Fore. Left-diag. RL Mean+ Fore. Hor. RL Mean+ Back. Left-diag. RL Variance+ Back. Ver. RL Variance + Fore. Right-diag. RL Mean	197234708	99.701	88211956	90.515	591696	0.299	9243478	9.485	96.669	3.331
Fore. Left-diag. RL Mean+ Fore. Hor. RL Mean+ Back. Left-diag. RL Variance+ Back. Ver. RL Variance + Background Mean	195515024	98.832	87950559	90.247	2311380	1.168	9504875	9.753	95.998	4.002
Fore. Left-diag. RL Mean+ Fore. Hor. RL Mean+ Back. Left-diag. RL Variance+ Back. Ver. RL Variance + Fore. Right-diag. RL Mean+ Back. Right-diag. RL Variance	192596645	97.356	92648479	95.068	5229759	2.644	4806955	4.932	96.601	3.399
Fore. Left-diag. RL Mean+ Fore. Hor. RL Mean+ Back. Left-diag. RL Variance+ Back. Ver. RL Variance + Fore. Right-diag. RL Mean+ Fore. Ver. RL Mean	185684075	93.862	90213631	92.569	12142329	6.138	7241803	7.431	93.435	6.565
Fore. Left-diag. RL Mean+ Fore. Hor. RL Mean+ Back. Left-diag. RL Variance+ Back. Ver. RL Variance + Fore. Right-diag. RL Mean+ Back. Right-diag. RL Variance+ Foreground Mean	192463653	97.289	93155931	95.588	5362751	2.711	4299503	4.412	96.728	3.272
Fore. Left-diag. RL Mean+ Fore. Hor. RL Mean+ Back. Left-diag. RL Variance+ Back. Ver. RL Variance + Fore. Right-diag. RL Mean+ Back. Right-diag. RL Variance+ Foreground Mean+ Fore. Ver. RL Mean	**197299465**	**99.734**	**92543599**	**94.960**	**526939**	**0.266**	**4911835**	**5.040**	**98.158**	**1.842**

Fore. Left-diag. RL Mean+ Fore. Hor. RL Mean+ Back. Left-diag. RL Variance+ Back. Ver. RL Variance + Fore. Right-diag. RL Mean+ Back. Right-diag. RL Variance+ Foreground Mean+ Fore. Ver. RL Mean+ Background Mean	181612727	91.804	86986182	89.257	16213677	8.196	10469252	10.743	90.964	9.036

Table 15: The results of the features selection on the proposed approach based on SBFS method using the NN classifier

Features Names	Correct Classification				Miss classification				Accuracy %	Error %
	Text / Text		Non Text / Non Text		Text / Non Text		Non Text / Text			
	Area	%	Area	%	Area	%	Area	%		
Stand. Dev. of fore.+ # of Con. Comp.+ Background Mean+ Back. Hor. RL Mean+ Back. Hor. RL Variance+ Back. Ver. RL. Mean+ Back. Ver. RL Variance+ Back. Left-diag. RL Mean+ Back. Left-diag. RL Variance+ Back. Right-diag. RL Mean+ Back. Right-diag. RL Variance	**8466107**	**98.944**	**7274361**	**97.254**	**90363**	**1.056**	**205419**	**2.746**	**98.156**	**1.844**
Stand. Dev. of fore.+ # of Con. Comp.+ Background Mean+ Back. Hor. RL Mean+ Back. Hor. RL Variance+ Back. Ver. RL Variance+ Back. Left-diag. RL Mean+ Back. Left-diag. RL Variance+ Back. Right-diag. RL Mean+ Back. Right-diag. RL Variance	8459081	98.862	7274002	97.249	97389	1.138	205778	2.751	98.109	1.891
Stand. Dev. of fore.+ # of Con. Comp.+ Background Mean+ Back. Hor. RL Mean+ Back. Hor. RL Variance+ Back. Ver. RL Variance+ Back. Left-diag. RL Mean+ Back. Right-diag. RL Mean+ Back. Right-diag. RL Variance	8454309	98.806	7283104	97.371	102161	1.194	196676	2.629	98.136	1.864
Stand. Dev. of fore.+ # of Con. Comp.+ Back. Hor. RL Mean+ Back. Hor. RL Variance+ Back. Ver. RL Variance+ Back. Left-diag. RL Mean+ Back. Right-diag. RL Mean+ Back. Right-diag. RL Variance	8464316	98.923	7221172	96.543	92154	1.077	258608	3.457	97.813	2.187
Stand. Dev. of fore.+ # of Con. Comp.+ Background Mean+ Back. Hor. RL Mean+ Back. Hor. RL Variance+ Back. Left-diag. RL Mean+ Back. Right-diag. RL Mean+ Back. Right-diag. RL Variance	8455027	98.814	7204295	96.317	101443	1.186	275485	3.683	97.650	2.350
# of Con. Comp.+ Back. Hor. RL Mean+ Back. Hor. RL Variance+ Back. Ver. RL Variance+ Back. Left-diag. RL Mean+ Back. Right-diag. RL Mean+ Back. Right-diag. RL Variance	8460343	98.877	7272523	97.229	96127	1.123	207257	2.771	98.108	1.892

Stand. Dev. of fore.+ # of Con. Comp.+ Back. Hor. RL Mean+ Back. Hor. RL Variance+ Back. Left-diag. RL Mean+ Back. Right-diag. RL Mean+ Back. Right-diag. RL Variance	8463582	98.914	7232833	96.698	92888	1.086	246947	3.302	97.881	2.119
# of Con. Comp.+ Back. Hor. RL Mean+ Back. Hor. RL Variance+ Back. Ver. RL Variance+ Back. Right-diag. RL Mean+ Back. Right-diag. RL Variance	8464287	98.923	7250994	96.941	92183	1.077	228786	3.059	97.998	2.002
# of Con. Comp + Back. Hor. RL Variance+ Back. Ver. RL Variance+ Back. Right-diag. RL Mean+ Back. Right-diag. RL Variance	8470876	99.000	7255657	97.004	85594	1.000	224123	2.996	98.069	1.931
# of Con. Comp + Back. Hor. RL Variance+ Back. Right-diag. RL Mean+ Back. Right-diag. RL Variance	8483850	99.151	7235201	96.730	72620	0.849	244579	3.270	98.022	1.978
# of Con. Comp + Back. Hor. RL Variance+ Back. Ver. RL Variance+ Back. Right-diag. RL Mean	8488314	99.203	7187164	96.088	68156	0.797	292616	3.912	97.750	2.250
# of Con. Comp + Back. Hor. RL Variance+ Back. Right-diag. RL Mean	8488715	99.208	7187037	96.086	67755	0.792	292743	3.914	97.752	2.248
# of Con. Comp+ Back. Right-diag. RL Mean+ Back. Right-diag. RL Variance	8488819	99.209	7069566	94.516	67651	0.791	410214	5.484	97.020	2.980
# of Con. Comp+ Back. Right-diag. RL Mean	8488619	99.207	7189689	96.122	67851	0.793	290091	3.878	97.768	2.232
Back. Right-diag. RL Mean	7669156	89.630	7262536	97.096	887314	10.370	217244	2.904	93.112	6.888
# of Con. Comp	8488069	99.201	5632926	75.309	68401	0.799	1846854	24.691	88.057	11.943

Table 16: The results of the features selection on the XY cut approach based on SBFS method using the NN classifier

Feature	Correct Classification				Miss classification				Accuracy %	Error %
	Text / Text		Non Text / Non Text		Text / Non Text		Non Text / Text			
	Area	%	Area	%	Area	%	Area	%		
Mean of Hor. Proj.+ Fore. Hor. RL Mean+ Fore. Hor. RL Variance+ Fore. Ver. RL Mean+ Fore. Ver. RL Variance+ Back. Ver. RL Variance+ Fore. Left-diag. RL Mean+ Fore. Left-diag. RL Variance+ Back. Left-diag. RL Variance+ Fore. Right-diag. RL Variance+ Back. Right-diag. RL Variance	168910064	99.703	50918164	94.223	503058	0.297	3122092	5.777	98.378	1.622
Mean of Hor. Proj.+ Fore. Hor. RL Mean+ Fore. Hor. RL Variance+ Fore. Ver. RL Mean+ Fore. Ver. RL Variance+ Fore. Left-diag. RL Mean+ Fore. Left-diag. RL Variance+ Back. Left-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Right-diag. RL Variance+ Back. Right-diag. RL Variance	168783714	99.628	50129717	92.764	629408	0.372	3910539	7.236	97.968	2.032

Mean of Hor. Proj.+ Fore. Hor. RL Mean+ Fore. Hor. RL Variance+ Fore. Ver. RL Mean+ Fore. Ver. RL Variance+ Back. Ver. RL Variance+ Fore. Left-diag. RL Variance+ Back. Left-diag. RL Variance+ Fore. Right-diag. RL Variance+ Back. Right-diag. RL Variance	167965350	99.145	50129717	92.764	1447772	0.855	3910539	7.236	97.602	2.398
Mean of Hor. Proj.+ Fore. Hor. RL Mean+ Fore. Hor. RL Variance+ Fore. Ver. RL Mean+ Fore. Ver. RL Variance+ Back. Ver. RL Variance+ Fore. Left-diag. RL Mean+ Back. Left-diag. RL Variance+ Fore. Right-diag. RL Variance+ Back. Right-diag. RL Variance	167386732	98.804	50068686	92.651	2026390	1.196	3971570	7.349	97.316	2.684
Mean of Hor. Proj.+ Fore. Hor. RL Mean+ Fore. Hor. RL Variance+ Fore. Ver. RL Mean+ Fore. Ver. RL Variance+ Back. Ver. RL Variance+ Back. Left-diag. RL Variance+ Fore. Right-diag. RL Variance+ Back. Right-diag. RL Variance	167613160	98.938	51922345	96.081	1799962	1.062	2117911	3.919	98.247	1.753
Fore. Hor. RL Mean+ Fore. Hor. RL Variance+ Fore. Ver. RL Mean+ Fore. Ver. RL Variance+ Back. Ver. RL Variance+ Fore. Left-diag. RL Variance+ Back. Left-diag. RL Variance+ Fore. Right-diag. RL Variance+ Back. Right-diag. RL Variance	168907343	99.701	50322781	93.121	505779	0.299	3717475	6.879	98.110	1.890
Mean of Hor. Proj.+ Fore. Hor. RL Mean+ Fore. Hor. RL Variance+ Fore. Ver. RL Mean+ Fore. Ver. RL Variance+ Back. Ver. RL Variance+ Back. Left-diag. RL Variance+ Fore. Right-diag. RL Variance	168910064	99.703	50297381	93.074	503058	0.297	3742875	6.926	98.100	1.900
Fore. Hor. RL Mean+ Fore. Hor. RL Variance+ Fore. Ver. RL Mean+ Fore. Ver. RL Variance+ Back. Ver. RL Variance+ Back. Left-diag. RL Variance+ Fore. Right-diag. RL Variance	168834839	99.659	49056840	90.778	578283	0.341	4983416	9.222	97.511	2.489
Fore. Hor. RL Mean+ Fore. Hor. RL Variance+ Fore. Ver. RL Mean+ Back. Ver. RL Variance+ Back. Left-diag. RL Variance+ Fore. Right-diag. RL Variance	168892007	99.692	50708024	93.834	521115	0.308	3332232	6.166	98.276	1.724
Fore. Hor. RL Mean+ Fore. Hor. RL Variance+ Fore. Ver. RL Mean+ Back. Ver. RL Variance+ Fore. Right-diag. RL Variance	**168892007**	**99.692**	**51787615**	**95.832**	**521115**	**0.308**	**2252641**	**4.168**	**98.759**	**1.241**
Fore. Hor. RL Mean+ Fore. Hor. RL Variance+ Fore. Ver. RL Mean+ Back. Ver. RL Variance	168918569	99.708	49432799	91.474	494553	0.292	4607457	8.526	97.717	2.283
Fore. Hor. RL Mean+ Fore. Hor. RL Variance+ Back. Ver. RL Variance	168923383	99.711	48880634	90.452	489739	0.289	5159622	9.548	97.472	2.528
Fore. Hor. RL Mean+ Back. Ver. RL Variance	168732669	99.59	49064014	90.792	680453	0.402	4976242	9.208	97.469	2.531
Fore. Hor. RL Mean	168732669	99.598	38807810	71.813	680453	0.402	15232446	28.187	92.879	7.121
Back. Ver. RL Variance	169397786	99.991	29109160	53.866	15336	0.009	24931096	46.134	88.836	11.164

Table 17: The results of the features selection on the RLSA approach based on SBFS method using the NN classifier

Features Names	Correct Classification				Miss classification				Accuracy %	Error %
	Text / Text		Non Text / Non Text		Text / Non Text		Non Text / Text			
	Area	%	Area	%	Area	%	Area	%		
Foreground Mean+ Background Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Mean+ Back. Ver. RL Variance+ Fore. Left-diag. RL Mean+ Back. Left-diag. RL Variance+ Back. Right-diag. RL Variance	186536374	94.293	93588612	96.032	11290030	5.707	3866822	3.968	94.867	5.133
Foreground Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Mean+ Back. Ver. RL Variance+ Fore. Left-diag. RL Mean+ Back. Left-diag. RL Variance+ Back. Right-diag. RL Mean+ Back. Right-diag. RL Variance	183128370	92.570	92889839	95.315	14698034	7.430	4565595	4.685	93.476	6.524
Foreground Mean+ Background Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Mean+ Back. Ver. RL Variance+ Back. Left-diag. RL Variance+ Back. Right-diag. RL Variance	192339382	97.226	93735204	96.183	5487022	2.774	3720230	3.817	96.882	3.118
Foreground Mean+ Background Mean+ Fore. Hor. RL Mean+ Back. Ver. RL Variance+ Fore. Left-diag. RL Mean+ Back. Left-diag. RL Variance+ Back. Right-diag. RL Variance	190295125	96.193	92736606	95.158	7531279	3.807	4718828	4.842	95.851	4.149
Foreground Mean+ Background Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Mean+ Back. Left-diag. RL Variance+ Back. Right-diag. RL Variance	186901976	94.478	94210825	96.671	10924428	5.522	3244609	3.329	95.202	4.798
Foreground Mean+ Background Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Mean+ Back. Left-diag. RL Variance+ Back. Right-diag. RL Variance	187170600	94.614	93365183	95.803	10655804	5.386	4090251	4.197	95.006	4.994
Background Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Mean+ Back. Left-diag. RL Variance+ Back. Right-diag. RL Variance	**193463898**	**97.795**	**92801140**	**95.224**	**4362506**	**2.205**	**4654294**	**4.776**	**96.946**	**3.054**
Foreground Mean+ Background Mean+ Fore. Hor. RL Mean+ Back. Left-diag. RL Variance+ Back. Right-diag. RL Variance	191083724	96.592	93105390	95.536	6742680	3.408	4350044	4.464	96.243	3.757
Background Mean+ Fore. Hor. RL Mean+ Back. Left-diag. RL Variance+ Back. Right-diag. RL Variance	187754100	94.909	93521722	95.964	10072304	5.091	3933712	4.036	95.257	4.743
Background Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Mean+ Back. Right-diag. RL Variance	185355678	93.696	93749946	96.198	12470726	6.304	3705488	3.802	94.522	5.478

Fore. Hor. RL Mean+ Back. Left-diag. RL Variance+ Back. Right-diag. RL Variance	193430250	97.778	88410871	90.719	4396154	2.222	9044563	9.281	95.448	4.552
Background Mean+ Fore. Hor. RL Mean+ Back. Right-diag. RL Variance	187989778	95.028	90580293	92.945	9836626	4.972	6875141	7.055	94.340	5.660
Fore. Hor. RL Mean+ Back. Right-diag. RL Variance	195406170	98.777	87035997	89.309	2420234	1.223	10419437	10.691	95.652	4.348
Fore. Hor. RL Mean+ Back. Left-diag. RL Variance	193262379	97.693	85407285	87.637	4564025	2.307	12048149	12.363	94.374	5.626
Fore. Hor. RL Mean	195116797	98.630	83902932	86.094	2709607	1.370	13552502	13.906	94.493	5.507
Back. Right-diag. RL Variance	195705176	98.928	64707778	66.397	2121228	1.072	32747656	33.603	88.191	11.809

Table 18: The results of using K-NN and SVM classifiers to evaluate the best features selected by SFFS

The algorithm	Features Names	Classifier	Correct Classification				Misclassification				Accuracy %	Error %
			Text / Text		Non Text / Non Text		Text / Non Text		Non Text / Text			
			Area	%	Area	%	Area	%	Area	%		
Proposed Segmentation Algorithm	Back. Ver. RL. Mean + # of Con. Comp+ Back. Left-diag. RL Mean + Back. Hor. RL Variance + Background Mean+ Back. Left-diag. RL Variance+ Back. Hor. RL Mean+ Back. Right-diag. RL Mean	K-NN	8430488	98.528	7161493	95.769	125982	1.472	316359	4.231	97.241	2.759
		SVM	8512640	99.488	6562668	87.761	43830	0.512	915184	12.239	94.019	5.981
XY Cut	Fore. Right-diag. RL Variance+ Fore. Right-diag. RL Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Variance + Back. Right-diag. RL Variance+ Fore. Left-diag. RL Variance+ Back. Left-diag. RL Variance+ Fore. Hor. RL Variance+ Back. Ver. RL Variance	K-NN	164857756	97.305	38500280	71.245	4565954	2.695	15539216	28.755	91.003	8.997
		SVM	168928676	99.708	38577284	71.387	495034	0.292	15462212	28.613	92.859	7.141
RLSA	Fore. Left-diag. RL Mean+ Fore. Hor. RL Mean+ Back. Left-diag. RL Variance+ Back. Ver. RL Variance + Fore. Right-diag. RL Mean+ Back. Right-diag. RL Variance+ Foreground Mean+ Fore. Ver. RL Mean	K-NN	187672056	94.863	89692871	92.022	10162702	5.137	7775786	7.9777	93.925	6.075
		SVM	185121000	93.574	88633361	90.935	12713758	6.426	8835296	9.065	92.703	7.297

Table 19: The results of using K-NN and SVM classifiers to evaluate the best features selected by SBFS

The algorithm	Features Names	Classifier	Correct Classification				Misclassification				Accuracy %	Error %
			Text / Text		Non Text / Non Text		Text / Non Text		Non Text / Text			
			Area	%	Area	%	Area	%	Area	%		
Proposed Segmentation Algorithm	Stand. Dev. of fore.+ # of Con. Comp.+ Background Mean+ Back. Hor. RL Mean+ Back. Hor. RL Variance+ Back. Ver. RL. Mean+ Back. Ver. RL Variance+ Back. Left-diag. RL Mean+ Back. Left-diag. RL Variance+ Back. Right-diag. RL Mean+ Back. Right-diag. RL Variance	K-NN	8395346	98.117	7123897	95.267	161124	1.883	353955	4.733	96.788	3.212
		SVM	8512692	99.488	6542158	87.487	43778	0.512	935694	12.513	93.891	6.109
XY Cut	Fore. Hor. RL Mean+ Fore. Hor. RL Variance+ Fore. Ver. RL Mean+ Back. Ver. RL Variance+ Fore. Right-diag. RL Variance	K-NN	160004126	94.440	35531602	65.751	9419584	5.560	18507894	34.249	87.502	12.498
		SVM	168729706	99.590	38741480	71.691	694004	0.410	15298016	28.309	92.844	7.156
RLSA	Background Mean+ Fore. Hor. RL Mean+ Fore. Ver. RL Mean+ Back. Left-diag. RL Variance+ Back. Right-diag. RL Variance	K-NN	188120999	95.090	91168908	93.537	9713759	4.910	6299749	6.463	94.577	5.423
		SVM	184620678	93.330	86533361	88.781	13214080	6.679	10935296	11.219	91.822	8.178

5.3.3. Script Identification:

The scripts are identified at the block and word levels. At the block level, which is described in section 4.2, the Gabor features are extracted from each block texture patch. The height of lines in each block texture patch is 30 pixels.

While at the word level, which is described in section 4.3, the Gabor features are extracted from word texture patch. The parameters of the word texture patch are as follows:

1. The height of each word in the patch is 75.
2. No horizontal or vertical spaces between the repeated word in the patch.

The classifiers, which are used to classify the scripts at region and word levels, are K-nearest neighbor (K-NN) with k=1, Nearest Mean (NM), Neural Network (NN) (Multilayer Perceptron- Back propagation), Support Vector Machine (SVM), Decision Tree, and Tree Boost. The Neural Network contains one hidden layer of six neurons. The transfer functions of the hidden layer and output layer are the Log-sigmoid and Linear transfer functions respectively. The SVM's kernel function is sigmoid function

At region level, the recognition rates, which are achieved with K-NN (K=1), Nearest Mean (NM), NN, SVM, Decision Tree, and Tree Boost classifiers, are 99.5238%, 94.8095%, 99.3571%, 99.5952%, 99.3095%, and 99.4762% respectively. They are shown in Table 20. The highest accuracy, which is 99.5952%, is achieved with the SVM classifier. While the lowest accuracy, which is 94.8095%., is achieved with the Nearest Mean (NM) classifier.

At word level, the recognition rates, which are achieved with K-NN (K=1), Nearest Mean (NM), NN, SVM, Decision Tree, and Tree Boost classifiers, are 99.51%, 83.68%, 99.29%, 99.76%, 99.3%, and 99.59% respectively. They are shown in Table 21. SVM classifier shows the highest accuracy, which is 99.76%. While Nearest Mean (NM) classifier give the lowest accuracy, which is 83.68%.

The SVM and Tree Boost classifiers give better accuracies at the words level than the region level. Nearest Mean (NM) gives better accuracy at region level than word level. While the K-NN (K=1), NN, and Decision Tree classifiers give better accuracies at the region level than the word level with very small difference.

All classifiers provide very high accuracy at the region and word levels except the Nearest Mean (NM) classifier. The very high accuracy at word level enables us to identify the script at the word level.

At word level, about 30% of misclassified samples when using K-NN have special characters or numbers, 6.12% are due to skewed images, and 2.04% are due to Latin samples that look like Arabic or Arabic samples that look like Latin. When using Nearest Mean, 6.62% of misclassified samples have special characters or numbers, and about 6.37% are due to Latin samples that look like Arabic or Arabic samples that look like Latin. When using Neural Network, 9.86% of misclassified samples have special characters or numbers, 2.82% are due to skewed images, and about 7.04% are due to Latin samples that look like Arabic or Arabic samples that look like Latin. When using SVM, 20.83% of misclassified samples have special characters or numbers, and about 29.17% are due to Latin samples that look like Arabic or Arabic samples that look like Latin. When using Decision Tree classifier, 8.57% of

misclassified samples have special characters or numbers, 2.86% are unclear images, 10.00% are due to skewed images, and about 12.86% are due to Latin samples that look like Arabic or Arabic samples that look like Latin. When using Tree Boost classifier, 9.76% of misclassified samples have special characters or numbers, 4.88% are unclear images, 7.32% are due to skewed images, and about 12.20% are Latin samples that look like Arabic or Arabic samples that look like Latin. Figure 22 shows some of the misclassified samples at the word level.

At the region level, most misclassified samples has small word repeated, bad images, or skewed images. Figure 23 shows some of the misclassified samples.

Table 20: The Result of script identification at the block level using K- NN with K=1, NM, NN, SVM, Decision Tree, and Tree Boost classifiers

K- Nearest Neighbor (K-NN) with K=1				
Ground Truth	Recognition			
	Arabic	Latin	Accuracy	Error
Arabic	2085	15	99.2857%	0.7143%
Latin	5	2095	99.7619%	0.2381%
Average			99.5238%	0.4762%

Nearest Mean (NM)				
Ground Truth	Recognition			
	Arabic	Latin	Accuracy	Error
Arabic	1979	121	94.2381%	5.7619%
Latin	97	2003	95.3809%	4.6190%
Average			94.8095%	5.1905%

Neural Network (NN)				
Ground Truth	Recognition			
	Arabic	Latin	Accuracy	Error
Arabic	2092	8	99.619%	0.381%
Latin	19	2081	99.0952%	0.9048%
Average			99.3571%	0.6429%

Support Vector Machine (SVM)				
Ground Truth	Recognition			
	Arabic	Latin	Accuracy	Error
Arabic	2093	7	99.6667%	0.3333%
Latin	10	2090	99.5238%	0.4762%
Average			**99.5952%**	**0.4048%**

Decision Tree				
Ground Truth	Recognition			
	Arabic	Latin	Accuracy	Error
Arabic	2083	17	99.1905%	0.8095%
Latin	12	2088	99.4286%	0.5714%
Average			99.3095%	0.6905%

Tree Boost				
Ground Truth	Recognition			
	Arabic	Latin	Accuracy	Error
Arabic	2090	10	99.5238%	0.4762%
Latin	12	2088	99.4286%	0.5714%
Average			99.4762%	0.5238%

Table 21: The Result of script identification at the word level using K- NN with K=1, NM, NN, SVM, Decision Tree, and Tree Boost classifiers

K- Nearest Neighbor (K-NN) with K=1				
Ground Truth	Recognition			
	Arabic	Latin	Accuracy	Error
Arabic	4991	9	99.82%	0.18%
Latin	40	4960	99.2%	0.8%
Average			99.51%	0.49%

Nearest Mean (NM)				
Ground Truth	Recognition			
	Arabic	Latin	Accuracy	Error
Arabic	3985	1015	79.700%	20.300%
Latin	617	4383	87.660%	12.340%
Average			83.680%	16.320%

Neural Network (NN)				
Ground Truth	Recognition			
	Arabic	Latin	Accuracy	Error
Arabic	4974	26	99.480%	0.520%
Latin	45	4955	99.100%	0.900%
Average			99.290%	0.710%

Support Vector Machine (SVM)				
Ground Truth	Recognition			
	Arabic	Latin	Accuracy	Error
Arabic	4987	13	99.740%	0.260%
Latin	11	4989	99.780%	0.220%
Average			**99.76%**	**0.240%**

Decision Tree				
Ground Truth	Recognition			
	Arabic	Latin	Accuracy	Error
Arabic	4957	43	99.14%	0.860%
Latin	27	4973	99.46%	0.540%
Average			99.3%	0.700%

Tree Boost				
Ground Truth	Recognition			
	Arabic	Latin	Accuracy	Error
Arabic	4975	25	99.5%	0.500%
Latin	16	4984	99.68%	0.320%
Average			99.59%	0.410%

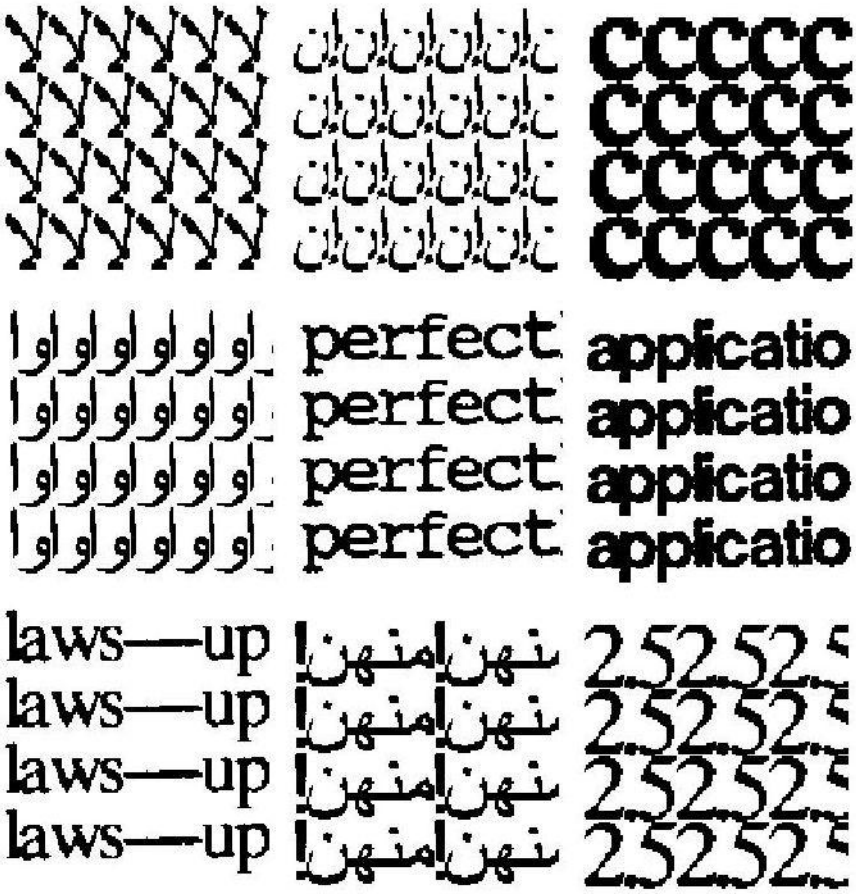


Figure 22: Some Misclassified Samples at word level

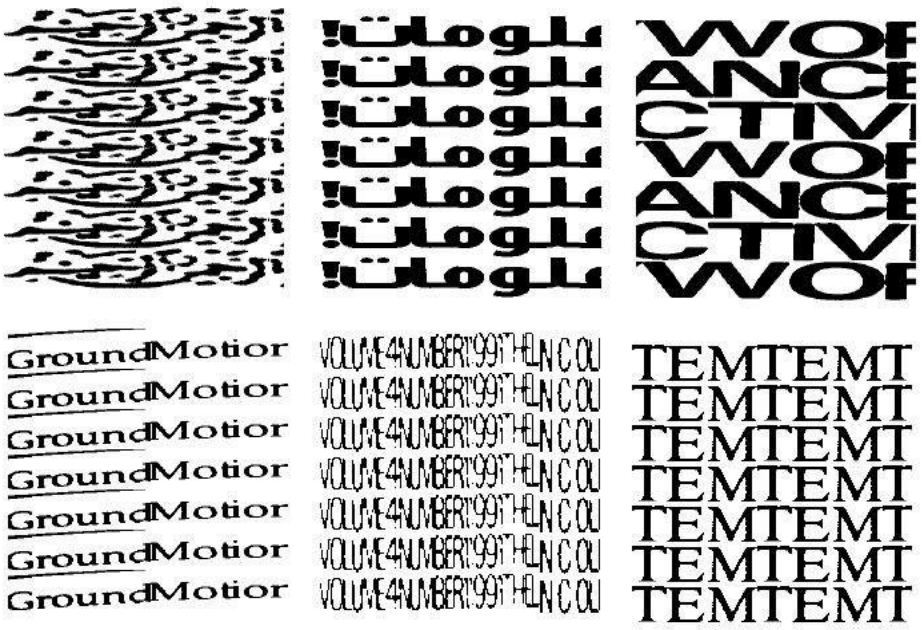


Figure 23: Some Misclassified Samples at region level

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this chapter, our conclusions are summarized in Section 6.1, and suggestions for future work are indicated in Section 6.2.

6.1. Conclusions

In this thesis, we have presented a prototype to segment a document image into text and non text regions, then classified text regions as Arabic or Latin scripts. We divided the system into two parts, namely document analysis and classification, and script identification. Document analysis and classification contains four phases: preprocessing, document segmentation, feature extraction, and document classification. The script identification has three steps, namely generate texture patches, texture feature extraction, script classification.

Preprocessing is necessary to enhance document images by binarization, noise removal, and skew detection and correction. For binarization, the Otsu algorithm [Otsu79] was used to identify the threshold used for converting a gray-level image into binary image. For noise removal, we used the Statistical Based Smoothing algorithm [Mahm94]. The Statistical Based Smoothing algorithm eliminates or fills pixels of the image based on its initial value and its neighbors' initial values.

In Document segmentation, a proposed segmentation algorithm was presented to segment documents into homogenous regions. The proposed segmentation algorithm consists of

three steps, namely rescaling the image, finding the boundaries of foreground pixels, and assigning regions. In rescaling the image, the image is divided into a number of n*n pixel windows to produce a scaled image. In the scaled image, each window is represented as one background pixel (white) if the number of black pixels in the window are less than a threshold otherwise as foreground (black). In finding the boundaries of foreground pixels, the boundaries of each connected component of the scaled image are found, then each connected component is assigned as a region. To evaluate the proposed algorithm, we used two types of region-based error measurements: merged and missed zones errors. In merged zones errors measure, the proposed algorithm has the best performance compared to XY cut and RLSA implemented algorithms. On the other hand, our proposed algorithm gives the worst performance in the missed zones measure as the rescaling steps that lead to loss of some pixels.

In document classification, we used Neural Network (Multilayer Perceptron- Back propagation), K-nearest neighbor, and support vector machine to classify each region to text or non text based on a number of features extracted in feature extraction. We implemented a number of features. Some of the features are used by other researchers. For feature selection, we followed two feature selection methods, SFFS and SBFS. In SFFS, the features which achieved the highest accuracy in the proposed algorithm are the number of connected components, background mean, background horizontal run length mean, background horizontal run length variance, background vertical run length mean, background left-diagonal run length mean, background left-diagonal run length variance, and background right-diagonal run length mean. The highest achieved accuracy rate is 98.206%. While the accuracy of the best selected features using SBFS method is

98.156%. The selected features are the standard deviation of the foreground, number of connected components, background mean, background horizontal run length mean, background horizontal run length variance, background vertical run length mean, background vertical run length variance background left-diagonal run length mean, background left-diagonal run length variance, background right-diagonal run length mean, and background right-diagonal run length variance.

In general, the proposed algorithm shows the best performance in all cases except in the missed zones measure and recognition rate achieved with NN using SBFS method to select features. It gives lower accuracy than XY cut, but better than RLSA in recognition rate achieved with NN using SBFS method to select features. While it gives the worst performance in the missed zones measure due to loss of some pixels in the rescaling phase.

In script identification, features are extracted at the block and word levels. At the block level, each text region is divided into n*n block texture patches. The spaces between words/characters and text lines are removed. The height of its line is normalized to a fixed height. At the word level, word texture patch is generated in four steps. the First of all, all lines of a text region are extracted using horizontal projection. The words in each line are extracted using the vertical projection. The height of each word is normalized with a standard height. Finally, each word is repeated horizontally and vertically to fill a word texture patch, which has standard height and width.

In texture features extraction, Gabor features are extracted for each block and word texture patch (block and word levels). Then in script classification, we used K-nearest neighbor with K=1 (K-NN), nearest mean (NM), neural network (NN), support vector

machine (SVM), Decision Tree, and Tree Boost classifiers to classify the scripts of text regions as Arabic or Latin. At region level, the accuracy rates of KNN, NM, NN, SVM, Decision Tree, and Tree Boost classifiers are 99.5238%, 94.8095%, 99.357%, 99.5952%, 99.3095%, and 99.476% respectively. While at the word level, the accuracies , which are achieved with K-NN (K=1), NM, NN, SVM, Decision Tree, and Tree Boost classifiers, are 99.51%, 83.68%, 99.29%, 99.76%, 99.3%, and 99.59% respectively. Our experiments show high accuracy in most classifiers at region and word level.

6.2. Future Work

Some directions for further improving the performance of our system can be listed as follows:

1. The missed zone errors in the proposed segmentation algorithm need to be reduced. This error is due to missing some pixels in the rescaling the image step. As a future work, updating and enhancing the rules of the rescaled images may help in reducing most of the missed zone errors.
2. Extend the work for labeling text regions to title, abstract, footnote, caption or references.
3. Extend the work to label the non text regions to logos, forms, or etc.
4. In script identification, apply other types of texture features.

REFERENCES

[Otsu79] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions On Systems Man And Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[Mahm94] S. A. Mahmoud, "Pergamon Arabic Character Recognition Using Fourier Descriptors and Character Contour Encoding," *Pattern Recoonition*, vol. 27, no. 6, pp. 815–824, 1994.

[Bagd97] A. Bagdanov and J. Kanai, "Projection profile based skew estimation algorithm for JBIG compressed images," in *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, 1997, vol. 1, pp. 401–405 vol.1.

[Post86] W. Postl, "Detection of Linear Oblique Structures and Skew Scan in Digitized Documents," in *International Conference on Pattern Recognition*, 1986, pp. 687–689.

[Nicc99] G. Nicchiotti and C. Scagliola, "Generalised projections: a tool for cursive handwriting normalisation," in *Document Analysis and Recognition, 1999. ICDAR '99. Proceedings of the Fifth International Conference on*, 1999, pp. 729–732.

[Bair92] H. S. Baird, "Anatomy of a versatile page reader," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1059–1065, Jul. 1992.

[Gorm93] L. O. Gorman, "The document spectrum for page layout analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1162–1173, Nov. 1993.

[Hash86] A. Hashizume, P.-S. Yeh, and A. Rosenfeld, "A method of detecting the orientation of aligned components," *Pattern Recognition Letters*, vol. 4, no. 2, pp. 125–132, Apr. 1986.

[Chou07] C.-H. Chou, S.-Y. Chu, and F. Chang, "Estimation of skew angles for scanned documents based on piecewise covering by parallelograms," *Pattern Recognition*, vol. 40, pp. 443–455, Feb. 2007.

[YuJa96] B. Yu and A. K. Jain, "A robust and fast skew detection algorithm for generic documents," *Pattern Recognation*, vol. 29, no. 10, pp. 1599–1629, 1996.

[Manj07] V. N. Manjunath, G. H. Kumar, and P. Shivakumara, “Skew Detection Technique for Binary Document Images based on Hough Transform,” *International Journal of Information and Communication Engineering*, vol. 3, no. 7, pp. 493–499, 2007.

[Srih89] S. N. Srihari and V. Govindaraju, “Analysis of textual images using the Hough transform,” *Machine Vision and Applications*, vol. 2, no. 3, pp. 141–153, Jun. 1989.

[ChaC06] J. Cha, R. H. Cofer, and S. P. Kozaitis, “Extended Hough transform for linear feature detection,” *Pattern Recognition*, vol. 39, no. 6, pp. 1034–1043, Jun. 2006.

[Wong82] K. Y. Wong, R. G. Casey, and F. M. Wahl, “Document analysis system,” *IBM Journal of Research and Development*, vol. 26, pp. 647–656, 1982.

[Jaek95] H. Jaekyu, R. M. Haralick, and I. T. Phillips, “Recursive X-Y cut using bounding boxes of connected components,” in *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 2) - Volume 2 (ICDAR ’95)*, 1995, vol. 2, pp. 952–955.

[Nagy92] G. Nagy, S. Seth, and M. Viswanathan, “A prototype document image analysis system for technical journals,” *Computer*, vol. 25, no. 7, pp. 10–22, Jul. 1992.

[LiuT97] J. Liu, Y. Y. Tang, and C. Y. Suen, “Chinese document layout analysis based on adaptive split-and-merge and qualitative spatial reasoning,” *Pattern Recognition*, vol. 30, no. 8, pp. 1265–1278, Aug. 1997.

[Kise98] K. Kise, A. Sato, and M. Iwata, “Segmentation of Page Images Using the Area Voronoi Diagram,” *Computer Vision and Image Understanding*, vol. 70, no. 3, pp. 370–382, Jun. 1998.

[Hadj03] K. Hadjar and R. Ingold, “Arabic newspaper page segmentation,” in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings*, 2003, pp. 895– 899.

[ChiW03] Z. Chi, Q. Wang, and W. Siu, “Hierarchical content classiÿcation and script determination for automatic document image processing,” *Pattern Recognition*, vol. 36, pp. 2483–2500, 2003.

[Shir05] M. Shirali-shahreza and S. Shirali-shahreza, “A Robust Page Segmentation Method for Persian / Arabic Documents,” in *Proceedings of the 5th*

WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision, 2005, vol. 2005, pp. 163–169.

[Anto09] A. Antonacopoulos, S. Pletschacher, D. Bridson, and C. Papadopoulos, "ICDAR 2009 Page Segmentation Competition," in *10th International Conference on Document Analysis and Recognition, 2009. ICDAR '09*, 2009, pp. 1370–1374.

[Bair07] H. S. Baird, M. A. Moll, C. An, and M. R. Casey, "Document image content inventories," in *Proceedings of SPIE 6500, Document Recognition and Retrieval XIV, 65000X*, 2007.

[AnBa07] C. An, H. S. Baird, and P. Xiu, "Iterated document content classification," in *Ninth International Conference on Document Analysis and Recognition, 2007.ICDAR 2007*, 2007, pp. 252–256.

[Ingl95] S. Inglis and I. H. Witten, "Document Zone Classification Using Machine Learning," *Proceedings of Digital Image Computing: Techniques and Applications*, 1995.

[Sauv95] J. Sauvola and M. Pietikainen, "Page segmentation and classification using fast feature extraction and connectivity analysis," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1995, vol. 2, pp. 1127–1131.

[FanW95] K. C. Fan and L. S. Wang, "Classification of Document Blocks Using Density Feature and Connectivity Histogram," *Pattern Recognition Letters*, vol. 16, no. 9, pp. 955–962, 1995.

[Lian96] J. Liang, I. T. Phillips, J. Ha, and R. M. Haralick, "Document Zone Classification Using Sizes of Connected-components," in *Document Recognition III, SPIE'96*, 1996, pp. 150–157.

[Wang06] Y. Wang, I. T. Phillips, and R. M. Haralick, "Document zone content classification and its performance evaluation," *Pattern Recognition*, vol. 39, no. 1, pp. 57–73, Jan. 2006.

[Keys07] D. Keysers, F. Shafait, and T. M. Breuel, "Document image zone classification - a simple high-performance approach," in *Proceedings of second International Conference on Computer Vision Theory and Applications*, 2007, pp. 44–51.

[Dese04] T. Deselaers, D. Keysers, and H. Ney, "Features for Image Retrieval - A Quantitative Comparison," *In DAGM 2004, Pattern Recognition, 26th DAGM Symposium*, pp. 228–236, 2004.

[Okun99] O. Okun, D. Doermann, and M. Pietikainen, "Page Segmentation and Zone Classification: The State of the Art," in *Technical Report LAMP-TR-036, CAR-TR-927, CS-TR-4079, University of Maryland, College Park*, 1999.

[Bloo91] D. Bloomberg, "Multiresolution Morphological Approach to Document Image Analysis," in *International Conference on Document Analysis and Recognition*, 1991, pp. 963–971.

[Bukh11] S. S. Bukhari, F. Shafait, and T. M. Breuel, "High Performance Layout Analysis of Arabic and Urdu Document Images," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, 2011, pp. 1275–1279.

[Hoch97] J. Hochberg, P. Kelly, T. Thomas, and L. Kerns, "Automatic Script Identification From Document Images Using Cluster-Based Templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, vol. 19, no. 2, pp. 176–181, 1997.

[Spit97] A. L. Spitz, "Determination of the Script and Language Content of Document Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 235–245, 1997.

[Elga01] A. M. Elgammal and M. A. Ismail, "Techniques for Language Identification for Hybrid Arabic-English Document Images," in *Document Analysis and Recognition, Sixth International Conference*, 2001, pp. 1100–1104.

[Kano02] S. Kanoun, A. Ennaji, Y. Lecourtier, and A. M. Alimi, "Script and Nature Differentiation for Arabic and Latin Text Images," in *roceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, 2002, pp. 309– 313.

[Busc05] A. Busch, W. W. Boles, S. Sridharan, and S. Member, "Texture for Script Identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1720–1732, 2005.

[Jlai07] M. B. E. N. Jlaiel, S. Kanoun, A. M. Alimi, R. Mullot, and L. Rochelle, "Three decision levels strategy for Arabic and Latin texts differentiation in printed and handwritten natures," in *Proceedings of Ninth International*

Conference on Document Analysis and Recognition, 2007. ICDAR 2007., 2007, pp. 1103–1107.

[Mahm11] S. A. Mahmoud and W. G. Al-Khatib, "Recognition of Arabic (Indian) bank check digits using log-gabor filters," *Applied Intelligence*, vol. 35, no. 3, pp. 445–456, 2011.

[Alha09] A. G. AL-Hashim, "Arabic database for automatic printed Arabic text recognition research and benchmarking," MSc Thesis. KFUPM, Dhaharan, Saudi Arabia, 2009.

[Alha10a] A. G. Al-Hashim and S. A. Mahmoud, "Benchmark Database and GUI Environment for Printed Arabic Text Recognition Research," *WSEAS Transactions on Information Science and Applications*, vol. 7, no. 4, pp. 587–597, 2010.

[Alha10b] A. G. Al-Hashim and S. A. Mahmoud, "Printed Arabic Text Database (PATDB) for Research and Benchmarking," in *Proceedings of the 9th WSEAS international conference on Applications of Computer Engineering*, 2010, pp. 62–68.

[Phil93] I. . Phillips, S. Chen, and R. Haralick, "CD-ROM document database standard," in *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93)*, 1993, pp. 478–483.

[Shaf08] F. Shafait, D. Keysers, and T. M. Breuel, "Performance Evaluation and Benchmarking of Six-Page Segmentation Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 941–954, 2008.

[Sher03] P. H. Sherrod, "DTREG predictive modeling software," *Software available at http://www.dtreg.com*, 2003.

Vitae

Name	ZAHER AHMED SAHAL BAMASOOD
Nationality	Yemeni
Date of Birth	19th May, 1980
Email	zasb195@gmail.com
Address	Mukalla, Hadhramout, Yemen
Academic Background	Received B.S Computer Science from Hadhramout University of Science & Technology, Yemen in 2004. Joined King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia as a Master Student in February, 2009. Completed M.S in Computer Science from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia in October 2013.